Departament d'Enginyeria Telemàtica
UNIVERSITAT POLITÈCNICA DE CATALUNYA

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Thesis Advisor
Jorge Mata Díaz

Thesis Tutor
Juan José Alins Delgado

Contributions based on cross-layer design for Quality-of-Service provisioning over DVB-S2/RCS Broadband Satellite Systems

PhD Thesis

Elizabeth Rendón Morales

Barcelona, 2013

Departament d'Enginyeria Telemàtica
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Universitat Politècnica de Catalunya

Department of Telematics Engineering

Thesis Advisor

Jorge Mata Díaz
Associated Professor
Dr. Ing. de Telecomunicación
Dept. de Ingeniería Telematica
UPC Universitat Politècnica de Catalunya

Thesis Tutor

Juan José Alins Delgado
Assistant Professor
Dr. Ing. de Telecomunicación
Dept. de Ingeniería Telematica
UPC Universitat Politècnica de Catalunya

Contributions based on cross-layer design for Quality-of-Service provisioning over DVB-S2/RCS Broadband Satellite Systems

PhD Thesis

Thesis presented to obtain the title of Doctor
by the Universitat Politècnica de Catalunya

Elizabeth Rendón Morales
Researcher
Dept.de Ingeniería Telematica
UPC Universitat Politècnica de Catalunya
elizabeth.rendon@entel.upc.edu

Barcelona, 2013

Contributions based on cross-layer design for Quality-of-Service provisioning over DVB-S2/RCS Broadband Satellite Systems

Elizabeth Rendon Morales

PhD Thesis

Barcelona, 2013

With love to Rodrigo Aviles.

Abstract

Nowadays, geostationary (GEO) satellite infrastructure plays a crucial role for the provisioning of IP services. Such infrastructure can provide ubiquity and broadband access, being feasible to reach disperse populations located world wide within remote areas where terrestrial infrastructure can not be deployed.

Nevertheless, due to the expansion of the World Wide Web (WWW), new IP applications such as Voice over IP (VoIP) and multimedia services requires considering different levels of individual packet treatment through the satellite network. This differentiation must include not only the Quality of Service (QoS) parameters to specify packet transmission priorities across the network nodes, but also the required amount of bandwidth assignment to guarantee its transport.

In this context, the provisioning of QoS guarantees over GEO satellite systems becomes one of the main research areas of organizations such as the European Space Agency (ESA). Mainly because, their current infrastructures require continuous exploitation, as launching a new communication satellite is associated with excessive costs. Therefore, the support of IP services with QoS guarantees must be developed on the terrestrial segment to enable using the current assets.

In this PhD thesis several contributions to improve the QoS provisioning over DVB-S2/RCS Broadband Satellite Systems have been developed. The contributions are based on cross-layer design, following the layered model standardized in the ETSI TR 102 157 [ETS03] and 462 [10205]. The proposals take into account the drawbacks posed by GEO satellite systems such as delay, losses and bandwidth variations.

The *first contribution* proposes QoSatArt, an architecture defined to improve QoS provisioning among services classes considering the physical layer variations due to the presence of rain events. The design is developed inside the gateway, including the specification of the main functional blocks to provide QoS guarantees and mechanisms to minimize de delay and jitter values experienced at the application layer. Here, a cross-layer design between the physical and the network layer has been proposed, to enforce the QoS specifications based on the available bandwidth. The proposed QoSatArt architecture is evaluated using the NS-2 simulation tool. In addition, the performance analysis of several standard Transmission Control Protocol (TCP) variants is also performed. This is carry out to find the most suitable TCP variant that enhances TCP transmission over a QoS architecture such as the QoSatArt.

The *second contribution* proposes XPLIT, an architecture developed to enhance TCP transmission with QoS for DVB-S2/RCS satellite systems. Complementary to QoSatArt, XPLIT introduces Performance Enhanced Proxies (PEPs), which breaks the end-to-end semantic of TCP connections. However, it considers a cross-layer design between the network layer and the transport layer to enhance TCP transmission while providing them with QoS guarantees. Here, a modified TCP variant called XPLIT-TCP is proposed to send data through the forward and the return channel. XPLIT-TCP uses two control loops (the buffer occupancy and the service rate (β_i, μ_i)) to provide optimized congestion control functions. The proposed XPLIT architecture is evaluated using the NS-2 simulation tool.

Finally, the *third contribution* of this thesis consists on the development of a unified architecture to provide QoS guarantees based on cross-layer design over broadband satellite systems. It adopts the enhancements proposed by the QoSatArt architecture working at the network layer, in combination with the enhancements proposed by the XPLIT architecture working at the transport layer.

Resumen

Actualmente, los satélites Geoestacionarios (GEO) juegan un papel muy importante en la provisión de servicios IP. Esta infraestructura permite proveer ubicuidad y acceso de banda ancha, haciendo posible alcanzar poblaciones dispersas en zonas remotas donde la infraestructura terrestre es inexistente.

Sin embargo, en la provisión de aplicaciones como Voz sobre IP (VoIP) y servicios multimedia, es importante considerar el tratamiento diferenciado de paquetes a través de la red satelital. Esta diferenciación debe considerar no sólo los requerimientos de Calidad de Servicio (QoS) que especifican las prioridades de los paquetes a través de los nodos de red, si no también el ancho de banda asignado para garantizar su transporte.

En este contexto, la provisión de garantías de QoS sobre satélites GEO es una de las principales áreas de investigación de organizaciones como la Agencia Espacial Europea (ESA) persiguen. Esto se debe principalmente ya que dichas organizaciones requieren la explotación continua de sus activos, dado que lanzar un nuevo satélite al espacio representa costos excesivos. Como resultado, el soporte de servicios IP con calidad de servicio sobre la infraestructura satelital actual es de vital importancia.

En esta tesis doctoral se presentan varias contribuciones para el soporte a la Calidad de Servicio en redes DVB-S2/RCS satelitales de banda ancha. Las contribuciones propuestas se basan principalmente en el diseño "cross-layer" siguiendo el modelo de capas definido y estandarizado en las especificaciones ETSI TR 102 157 [ETS03] y 462 [10205]. Las contribuciones propuestas consideran las limitaciones presentes de los sistemas satelitales GEO como lo son el retardo de propagación, la perdida de paquetes y las variaciones de ancho de banda causados por eventos atmosféricos.

La *primera contribución* propone QoSatArt, una arquitectura definida para mejorar el soporte a la QoS. Esta arquitectura considera las variaciones en la capa física debido a la presencia de eventos de lluvia para priorizar los niveles de QoS. El diseño se desarrolla en el gateway e incluye las especificaciones de los principales elementos funcionales y mecanismos para garantizar la QoS y minimizar el retardo presente en la capa de aplicación. Aquí, se propone un diseño "cross-layer" entre la capa física y la capa de red, con el objetivo de reforzar las especificaciones de QoS considerando el ancho de banda disponible. La arquitectura QoSatArt es simulada y evaluada empleando la herramienta de simulación NS-2. Adicionalmente, un análisis de desempeño de diversas variantes de TCP (Transmission Control Protocol) es realizado con el objetivo de encontrar la variante de TCP más adecuada para trabajar en un ambiente con QoS como QoSatArt.

La *segunda contribución* propone XPLIT, una arquitectura desarrollada para mejorar las transmisiones TCP con QoS en un sistema satelital DVB-S2/RCS. Complementario a QoSatArt, XPLIT emplea PEPs (Performance Enhanced Proxies), afectando la semántica end-to-end de las conexiones TCP. Sin embargo, XPLIT considera un diseño "cross-layer" entre la capa de red y la capa de transporte con el objetivo de mejorar las transmisiones TCP considerando los parámetros de QoS como la ocupación de la cola y la tasa de transmisión (β_i, μ_i). Aquí, se propone el uso de una nueva variante de TCP es propuesta llamada XPLIT-TCP, que usa dos bucles para proveer funciones mejoradas en el control de congestión. La arquitectura XPLIT es simulada y evaluada empleando la herramienta de simulación NS-2.

Finalmente, la *tercera contribución* de esta tesis consiste en el desarrollo de un arquitectura unificada para el soporte a la QoS en redes satelitales de banda ancha basada en técnicas "cross-layer". Esta arquitectura adopta las mejoras propuestas por QoSatArt en la capa de red en combinación con las mejoras propuestas por XPLIT en la capa de transporte.

Acknowledgments

I would like to thank my supervisor Dr. Jorge Mata Diaz for his endless support and interest in the development of this thesis. Thank you for creating a nice working atmosphere, for being so patient, and for all the support and advice in each of the tasks in which I was involved during the development of this work.

I would also like to thank Dr. Juan Jose Alins, for his fruitful and valuable comments through the development of this work.

Personally, I would like to thank my Lord Jesús for being my cornerstone and the guide of my life.

Most of all, I am very grateful with my husband Rodrigo for his valuable support, thanks for being so positive, lovely and always motivate myself during the development of the doctorate. Also, I would like to specially thank my parents Pety and Heri, my brothers and sisters and also Rodrigo's family for their support.

Finally, I would like to thank the FPI-UPC grant, including the members of the department of Telematics Engineering and the Universitat Politècnica de Catalunya.

This work has been funded within the support of the the Spanish Ministry of Science and Education under the projects SERVET (TEC2011-26452-SERVET), CONSOLIDER-ARES (CSD2007-00004), ITACA (TSI2007-65393-C02-02) and P2PSEC (TEC 2008-06663-C03-01).

List of Publications

The presented thesis has been based on the work described in the following papers:

I. Elizabeth Rendon-Morales, Jorge Mata-Díaz, Juanjo Alins, Jose Muñoz, and Oscar Esparza. ***"Cross-layer packet scheduler for QoS support over DVB-S2 Broadband Satellite Systems"***. International Journal of Communication Systems, Wiley, 10/2012.

II. Elizabeth Rendon-Morales, Jorge Mata-Díaz, Juanjo Alins, Jose Muñoz, and Oscar Esparza. ***"QoSatAr: a cross-layer architecture for E2E QoS provisioning over DVB-S2 broadband satellite systems"***. EURASIP Journal on Wireless Communications and Networking, Springer, 09/2012.

III. Elizabeth Rendon-Morales, Jorge Mata-Díaz, Juanjo Alins, Jose Muñoz, and Oscar Esparza. ***"Adaptive IP scheduler design to support QoS guarantees over satellite systems"***. Journal of Internet Engineering, Klidarithmos Press, 08/2012.

IV. Elizabeth Rendon-Morales, Jorge Mata-Díaz, Juanjo Alins, Jose Muñoz, and Oscar Esparza. ***"Performance Evaluation of selected TCP variants over a DVB-S2 satellite system with QoS"***. International Journal of Communication Systems, Wiley, 03/2012.

V. Juanjo Alins, Jorge Mata-Díaz, Jose Muñoz, Elizabeth Rendon-Morales and Oscar Esparza. ***"XPLIT a Cross-layer architecture for TCP services over DVB-S2/ETSI QoS BSM"***. Computer Networks Journal, Elsevier, 03/2012.

VI. Elizabeth Rendon-Morales, Jorge Mata-Díaz, Juanjo Alins, Jose Muñoz, and Oscar Esparza. ***"XPLIT-RCS: a Cross-layer architecture for TCP services over DVB-RCS/ETSI QoS BSM"***. Submitted to International Journal of Satellite Communications and Networking, Wiley, 01/2013.

Additional relevant publications

- Elizabeth Rendón-Morales, Jorge Mata-Díaz, Juanjo Alins, Jose L. Munoz and Oscar Esparza ***"Cross-layer architecture for TCP splitting in the return channel over satellite networks"***. In Wireless Communication Systems, 2009. ISWCS 2009. 6th International Symposium on, pages 225 –229, sept. 2009. Technical Co-Sponsorship by IEEE Communications Society.

- Elizabeth Rendon-Morales, Jorge Mata-Díaz, Juanjo Alins, Jose Muñoz, and Oscar Esparza. ***"Adaptive packet scheduling for the support of QoS over DVB-S2 satellite systems"***. In Wired/Wireless Internet Communications, volume 6649 of Lecture Notes in Computer Science, pages 15–26. Springer Berlin / Heidelberg, 2011.

- Elizabeth Rendon-Morales, Jorge Mata-Díaz, Juanjo Alins, Jose Muñoz, and Oscar Esparza ***"Evaluación de prestaciones de diferentes variantes de TCP en un entorno satelital DVB-S2 "***. In X Jornadas de Ingeniería Telemática (JITEL), pages 18 – 22, sept. 2011.

- Elizabeth Rendon-Morales, Jorge Mata-Díaz, Juanjo Alins, Jose Muñoz, and Oscar Esparza ***"Analysis of TCP variants over a DVB-S2 satellite system with QoS"***. In Eighth ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks (ACM PE-WASUN), Nov. 2011.

- Elizabeth Rendon-Morales, Jorge Mata-Díaz, Juanjo Alins, Jose Muñoz, and Oscar Esparza ***"Satellite for Health: An Architecture to Provide E2E Quality of Service Guarantees"***. In the Forum on Ph.D. Research in Information and Communication Technologies, Nov. 2012.

Contents

List of Figures

List of Abbreviations

ACM	Adaptive Code and Modulation
ADSL	Asymmetric Digital Subscriber Line
AF	Assured Forwarding
AQM	Active Queue Management
APIs	Application Program Interfaces
BE	Best-Effort
BoD	Bandwidth on Demand
BDP	Bandwidth-Delay Product
BSM	Broadband Satellite Multimedia
CPE	Customer premises Equipment
CK	Complete Knowledge
CRA	Continuous Rate Assignment
CIR	Committed Information Rate
DAMA	Dynamic Assignment Multiple Access
DiffServ	Differentiated Services
DVB-S2	Digital Video Broadcasting-Second Generation
DVB-RCS	Digital Video Broadcasting-Return Channel Satellite
DT	Drop Tail
ESA	European Space Agency
E2E	End-to-End
ECN	Explicit Congestion Notification
EF	Expedited Forwarding
FCA	Free Capacity Assignment
GEO	Geostationary satellite
GSE	Generic Stream Encapsulation
IntServ	Integrated Services
IP	Internet Protocol
MIB	Management Information Base
ModCod	Modulation and Code
NGN	Next Generation Networks
NCC	Network Control Center
PEP	Performance Enhanced Proxies
PETRA	Performance Enhancing Transport Architecture
PHB	Per Hop Behaviour

PK	Partial Knowledge
PDS	Proportional Differentiated Service
QoS	Quality of Service
QoSatArt	QoS SAtellite ARchiTecture
RBDC	Rate-Based Dynamic Capacity
RED	Random Early Discard
RTO	Retransmission Time Out
RTt	Round Trip Time
RSVP	Resource Reservation Protocol
RQM	Re-Queuing Mechanism
SATIP6	Satellite Broadband Multimedia Systems for IPV6
SLS	Service Level Specification
SLA	Service Level Agreement
SD	Satellite-Dependent
SI	Satellite-Independent
SI-SAP	Satellite Independent- Service Access Point
SNMP	Single Network Management Protocol
SNIR	Signal-to-Noise-plus-Interference Ratio
SIP	Session Initiation Protocol
ST	Satellite Terminals
SCSP-TP	Space Communications Protocol-Transport Protocol
TCP	Transmission Control Protocol
TDM	Time Division Multiplexed
TBTP	Terminal Burst Time Plan
TB	Token Buckets
UDP	User Data Protocol
VoIP	Voice over IP
VBDC	Volume Based Dynamic Capacity
WWW	World Wide Web
WTP	Waiting Time Priority
WRR	Weighted Round Robin
XPLIT	Cross-Layer Architecture for TCP sPLITting

CHAPTER 1

Introduction

Nowadays, geostationary (GEO) satellite infrastructure plays a crucial role for the provision of IP services. Specially, because satellite systems can provide ubiquity and broadband access, being feasible to reach different users (i.e. individual clients, international enterprises and disperse populations). Such users can be allocated within remote areas where terrestrial infrastructure can not be deployed. In a global context, satellite communication systems are seen as the enabler for bridging the *Digital Divide*, being able to provide every user, independently of its geographical location, full access to the Information Society including the provisioning of IP services.

However, the provisioning of broadband satellite services critically depends on the availability of the satellite resources (i.e. radio spectrum). In this way, the Broadband Satellite Multimedia (BSM) working group has standardized in the ETSI TR 102 157 [ETS03] the concept of BSM system, which is defined to provide high-quality interactive multimedia communications to end-users, including the support of broadband Internet access. A generic ETSI-BSM system is shown on Fig. 1.1. As it is observed, it is composed of the following elements: At the space segment, one of more satellite aircrafts are set up (either in transparent or regenerative configuration). At the ground (also called terrestrial) segment, a Network Control Center (NCC), a gateway and several Satellite Terminals (STs) are set up. The satellite terminal is also defined to work in a joint configuration with the Customer premises Equipment (CPE) allocated at the end-user.

In addition, due to the expansion of the World Wide Web (WWW), the support of new IP applications such as Voice over IP (VoIP) and multimedia services requires considering different levels of individual packet treatment through the BSM system. This differentiation must include not only the Quality of Service (QoS) parameters to specify packet transmission priorities across the network nodes, but also the required amount of bandwidth assignment to guarantee its transport.

Therefore, the provisioning of QoS guarantees over GEO BSM satellite systems becomes one of the main objectives for organizations that manage satellite infrastructures in Europe such as the European Space Agency (ESA). Mainly because, such organization requires the continuous exploitation of their current assets, as launching a new commu-

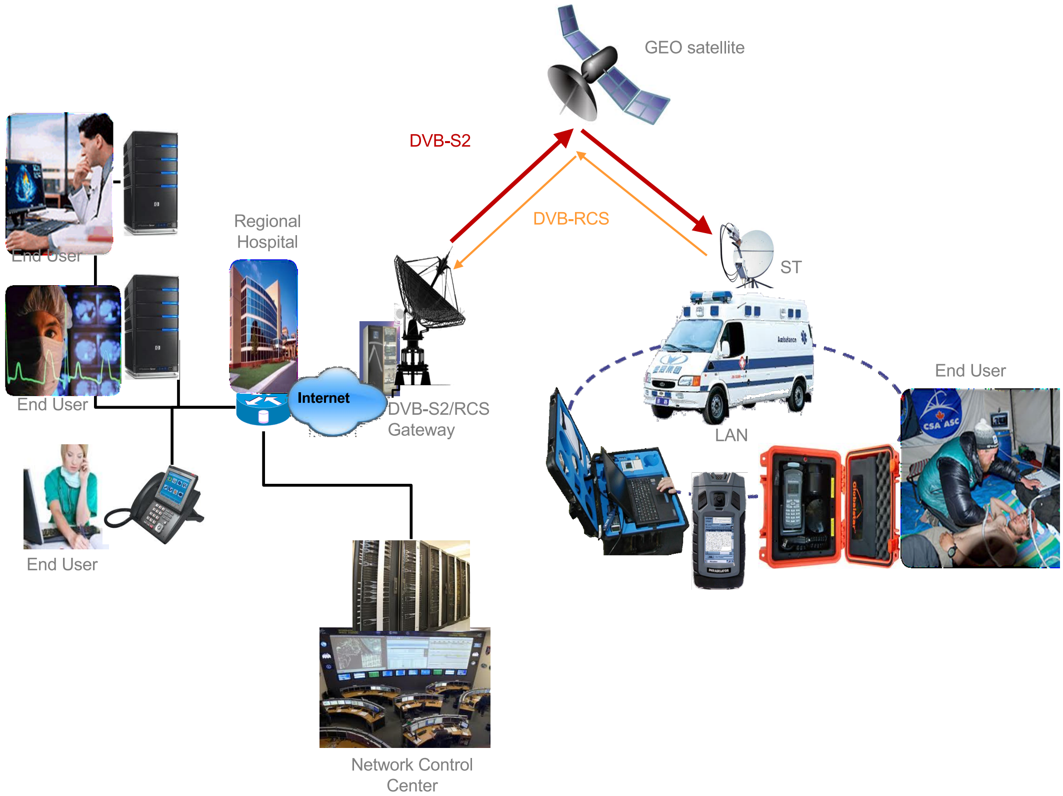


Figure 1.1: Generic ETSI-BSM system

nication satellite is associated with excessive costs. Therefore, the support of IP services with QoS guarantees over their current satellite communication systems become extremely important, taking advantage of the new proposals developed for the terrestrial segment.

Nevertheless, it is important to consider the satellite embedded characteristics that seriously affect the delivery of time critical data to end users. One of the main challenges that GEO satellite systems face is related to the available capacity or the available bandwidth. In particular, satellite systems working within the Ka frequency band (30/20 GHz) are able to provide higher bandwidths with smaller antennas (compared with Ku band). However, it presents the challenge of significant signal impairments in the presence of weather conditions such as rain events [Gia07]. Hence, if the available capacity is affected (reduced) by atmospheric events, it can be extended to a condition in which the satellite channel capacity is reduced, reaching its limit.

As a consequence, the greater the channel capacity is reduced, the lower the available bandwidth will be left to share among flows requiring QoS guarantees. Therefore, the differentiation of traffic becomes essential in order to guarantee the transmission of time critical data even though a reduced and limited channel capacity is experienced by the satellite system.

To address the challenges posed by the variability of the physical layer, the Digital Video Broadcasting-Second Generation/ Return Channel Satellite (DVB-S2/RCS) [ets04][ets05a] systems use the Channel Side Information (CSI) reported through the return channel by the satellite terminal in order to modify the transmission rates according to the channel conditions. This situation provides an increase of the system efficiency by taking advantage of the Signal-to-Noise-plus-Interference Ratio (SNIR) margin that is reserved for guaranteeing the service availability. In addition, the DVB-S2 specification defines as normative the use of the Adaptive Code and Modulation (ACM) [RVCM04] techniques to achieve quasi-error free channel conditions for each individual user, by providing them with the most suitable Modulation and Code (*ModCod*) value according to the measured SNIR. This ACM adaptation generates an important impact on the design of communications via satellite, as it is required to consider that the capacity in the satellite systems is constantly changing. Hence, a solution that addresses the issues related to the physical layer adaptability in order to provide QoS guarantees over the satellite path is required.

On the other hand, to address the challenges posed by the QoS provisioning, the BSM working group has also defined a functional architecture for the support of QoS guarantees over BSM systems. In particular, the ETSI-BSM architecture represents a layered model, which is characterized by the separation of two main layers named Satellite-Independent (SI) protocol layers and Satellite-Dependent (SD) layers [ETS05b]. This layered protocol architecture is shown on Fig. 1.2.

The interface between such layers is called Satellite Independent- Service Access Point (SI-SAP). In particular, the layers which their features depend on physical technology are called SD layers and can be associated with the physical and link layers of the OSI reference model. Alternatively, the layers which their characteristics are independent of the physical technology are called SI layers and can be associated with the network layer. In particular, the satellite SD layers are responsible for dealing with the resource utilization of the link while having an implicit knowledge of the wireless medium.

Nonetheless, to provide QoS differentiation, it is necessary to perform functions at the SI upper layers that the access scheme, at the SD layers, can not fully exploit. To address this, the support of sophisticated QoS methods over BSM satellite systems have been proposed and standardized in the ETSI TR 102 462 [10205] (called ETSI-BSM-QoS framework).

In this context, the SI layer plays a crucial role for the provisioning of QoS guarantees, given that at this layer, the establishment of priorities among users and applications that share the satellite link interface are defined. Here, the interaction between the higher layers and the lower layers becomes extremely important in order to encompass the service categorization (set at the higher layers) and the overall performance of the satellite network (set at the lower layers) [Mar07].

To address this, the use of a cross-layer design can be implemented to provide awareness between layers, enabling the interchange of information among layers [Gia07]. However,

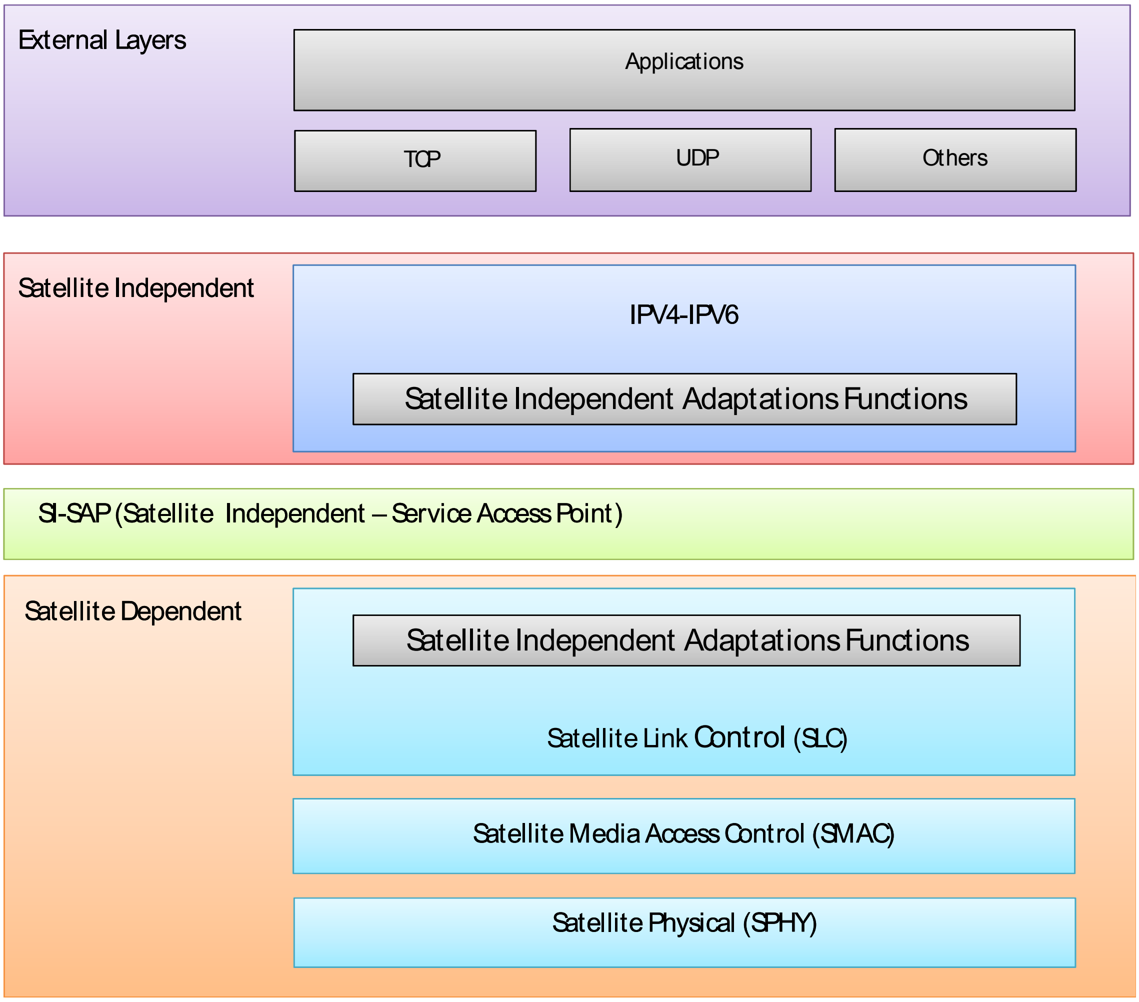


Figure 1.2: ETSI-BSM protocol architecture

the design of cross-layer mechanisms over satellite systems faces several challenges, as this design must pursue the minimization of the radio resources utilization, the optimization of the overall end-to-end application performance [BY04], while considering the QoS requirements for those applications requiring real time response. Moreover, the use of cross-layer techniques requires certain changes into the definition of protocols and systems while adding signaling mechanisms to enable the communication among protocol layers.

1.1 General Thesis Objective

In this thesis several contributions based on cross-layer design to enhance the provisioning of QoS guarantees over broadband satellite systems are proposed. The contributions are focused on the SI layers, following the specifications defined in the ETSI-BSM-QoS framework.

The design of such contributions becomes of particular interest, given that most of the related approaches propose the provisioning of QoS differentiation based on either the MAC layer (called *MAC-centric approach*) [CG07] or the application layer (called *Application-centric approach*) [GWE+09].

However, having a design focused on the SI layers (called *SI-centric approach*), it is possible to empower the management and control functions performed at the upper layers [M M08], while isolating the SD layers to include different physical layer supports.

The main driving force of the here presented thesis is to research into the interactions occurring among layers to have a complete design for providing QoS guarantees over DVB-S2/RCS satellite systems. This design takes into account the drawbacks posed by GEO satellite systems while implementing cross-layer techniques to provide QoS guarantees. Therefore, the main objective of this PhD thesis is to improve broadband satellite communications providing them with QoS support by means of a cross-layer design.

1.2 Thesis contributions

The thesis contributions can be classified considering the interactions among layers forming the ETSI-BSM protocol stack. The cross-layer interactions as a part of the thesis contributions are shown in Figure 1.3.

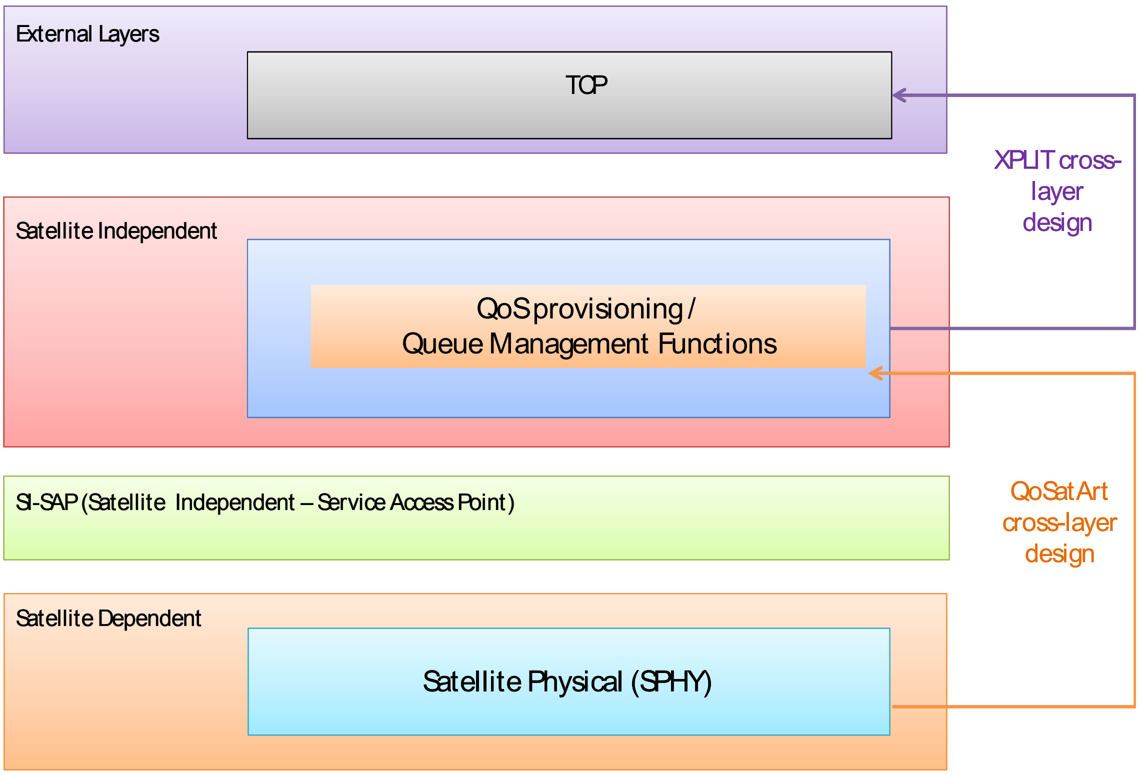


Figure 1.3: Cross-layer interactions as a part of the thesis contributions

In the presented thesis two main interactions have been researched: the first one analyzes the relationship between the SD lower layers and the SI network layer. Here, the first contribution of the thesis proposes the design of an architecture called QoSatArt, to provide QoS guarantees considering the physical layer adaptability. It includes the performance analysis of several standard TCP variants aiming to find the most suitable TCP variant that enhances TCP transmission over the proposed QoSatArt architecture.

The second interaction studies the relationship between the SI network layer and the transport layer (the External layer) to enforce QoS provisioning across the upper layers of the ETSI-BSM protocol stack. Within this interaction, the XPLIT architecture is proposed to enhance TCP transmission providing them with QoS guarantees. The XPLIT architecture is developed for the forward and the return channel, together with a modified TCP variant called XPLIT-TCP used to enhance TCP transmissions.

As it is observed on Fig. 1.3, the adoption of cross-layer techniques has been included to enhance the provisioning of QoS guarantees while looking for the optimization of the satellite resources. These contributions have been developed as a result of the oriented research work to obtain the doctorate degree, which are described as follows:

The first contribution of this research work proposes a complete QoS design developed for the DVB-S2 forward channel, which has been called QoSatArt (QoS SAtellite ARchiTecture) that considers the challenges posed by GEO satellite systems (i.e. delay, losses and bandwidth variations) while proposing several mechanisms based on a cross-layer design to enhance the provisioning of QoS guarantees.

The proposed architecture is based on the ETSI-BSM-QoS standards including several functional blocks to manage the QoS provisioning. Here, *a QoS design* inside the *gateway* has been detailed, including the design of *the Active Queue Management (AQM) system* that minimizes the delay values experienced at the Application layer for delay sensitive traffic. Moreover, a new mechanism has been developed in this work, which is called *Re-Queuing Mechanism (RQM).* This is used to enhance the goodput for the EF and AF traffic classes while reducing the end-to-end delay and jitter.

Together with this architecture, *a cross-layer IP scheduler*, allocated at the network layer has been also developed to guarantee the high priority traffic classes taking into account the channel condition affected by rain events. This mechanism is based on a cross-layer design that considers the current capacity of the DVB-S2 satellite link as the main criteria for the definition of QoS priorities, aiming to provide enhanced allocation of the satellite network resources. This cross-layer design is located between the physical layer and the network layer to enforce the QoS specifications defined in the SLA. In the view of the author, this is the first time a cross-layer optimization based on the SI network layer that uses the ACM adaptation behaviour as a main criteria to prioritize traffic classes over BSM satellite systems, has been proposed.

The natural step in the development of the present research work has been focused to efficiently address the QoS guarantees through the upper layers, particularly over the transport layer. As it is well know, transport protocols play a crucial role for the provision of End-to-End (E2E) services with QoS because these are closely related to the bandwidth efficiency in GEO satellite networks. Although the Transmission Control Protocol (TCP) is not the only protocol used to transport IP data, today, it is one of the most commonly used transport layer protocols on the Internet. However, given that the TCP protocol is severely affected by the long delay present in GEO satellite systems, it is clearly observed that even if lower layer based techniques are able to offer guaranteed bandwidth to TCP flows, the TCP protocol itself is not able to make an effective use of the network resources as a result of its intrinsic design. Here, the ideal solution can be the implementation of a transport layer able to adapt to the environmental characteristics in order to make more effective the provision of QoS guarantees.

In this way, two basic approaches have been analyzed, the first one considers the possibility to directly adopt a selected transport protocol variant on an end-to-end (E2E) environment (trying to keep the E2E semantic) with QoS provisioning.

The second solution consists on studying the implementation of a modified TCP variant that considers several parameters to efficiently manage the TCP connections as well as dealing with QoS issues.

Based on such requirements, ***the second contribution*** of the presented thesis analyzes the possibility to directly adopt a selected transport protocol on an E2E environment (trying to keep the E2E semantic) with QoS provisioning. The designed environment is the QoSatArt architecture developed to provide QoS guarantees. In this thesis *an evaluation of several TCP variants* working over a DVB-S2 satellite system with QoS support is presented, in order to select the more suitable TCP variant to enhance the performance of the proposed QoSatArt architecture. In the view of the author, this is the first time that such analysis is performed considering an ETSI-BSM-QoS scenario.

On the other hand, ***the third contribution*** of this thesis is focused on developing a modified TCP variant based on cross-layer design to enhance the performance of the TCP protocol over the satellite segment. Therefore a new TCP variant called XPLIT-TCP is proposed, which has been developed considering the architecture design called XPLIT (Cross-Layer Architecture for TCP sPLITting). This architecture considers the challenges posed by GEO satellite systems (i.e. delay, losses and bandwidth variations) while proposing several mechanisms based on cross-layer design to enhance the performance of the TCP protocol. This architecture considers Performance Enhanced Proxies (PEPs) which breaks the end-to-end semantic of the TCP connections. However, the XPLIT architecture considers a cross-layer design between the network layer and the transport layer to enhance data transmission in the DVB-S2 forward satellite channel. It includes the design of a signaling mechanism aiming to control the cross-layer parameters between the PEPs and the gateways. The design is also applied to the return channel based on the DVB-RCS standard. It considers the Bandwidth on Demand (BoD) mechanism to dynamically assign the system resources.

Finally, ***the four contribution*** of this thesis consists on the development of an unified framework to provide QoS guarantees over satellite systems. This includes the enhancements adopted by the QoSatArt architecture working at the network layer, in combination with the enhancements proposed by the XPLIT architecture working at the transport layer.

1.3 Thesis Objectives

Based on the mentioned contributions, the main objectives that the present thesis pursues are described as follows:

- To design, develop and evaluate the QoSatArt architecture based on a cross-layer design to provide QoS guarantees over DVB-S2 considering the physical layer variability. To develop a complete design inside the gateway based on the ETSI-BSM-QoS standard, including the specification of the main functional blocks to provide QoS guarantees and enhancing mechanisms to minimize de delay and jitter values experienced at the application layer.

- To design and evaluate the cross-layer IP scheduler aiming to provide QoS differentiation considering the bandwidth availability present at the physical layer, which is affected by weather conditions such as rain events. To specify and model the scheduler algorithm to provide QoS differentiation among traffic flows.

- To analyze the interaction of several standard TCP variants to determine the most suitable TCP variant that can enhance TCP transmission over the QoSatArt architecture.

- To design, develop and evaluate the XPLIT architecture that considers intermediated gateways (PEPs) to modify transmission characteristics taking into account the QoS requirements defined at the network layer. To design the cross-layer mechanisms used to define the modified TCP variant called XPLIT-TCP, which is used to enhance TCP transmission.

- Finally, to develop an unified framework to provide QoS guarantees that integrate the enhancements developed for the QoSatArt architecture and the XPLIT architecture.

The presented research work is organized in 8 chapters: in chapter 2 the ETSI-BSM-QoS architecture is described, together with the state of the art focus on QoS provisioning. In chapter 3, the design of the QoSatArt architecture is developed, including the description of the main functional blocks to provide QoS guarantees. In chapter 4, the cross-layer IP scheduler design is detailed and the QoSatArt simulation results are presented. In chapter 5, the analysis of TCP variants using QoSatArt architecture is performed. In chapter 6, the design of XPLIT-TCP protocol together with the XPLIT architecture defined for the DVB-S2 forward channel are developed. In chapter 7, the design of the XPLIT architecture designed for the DVB-RCS return channel is presented. Finally, chapter 8 includes the design of an unified framework that adopts the enhancements proposed by the QoSatArt and XPLIT working together. In Chapter 9, the overall conclusions are stated.

CHAPTER 2

State of the Art

Satellite systems used for communications provide two major advantages which mainly lies in their wide coverage (hence the predominance of broadcast applications), which means they can reach areas that are uneconomic for terrestrial provisioning, and the ability to quickly install new services over the slower terrestrial roll out [ETC+11].

However, one of the main concerns when working with satellite communications systems is to make an efficient use of the satellite resources (i.e. power and spectrum allocation), as they are scarce and expensive. In addition, to support different applications such as data, video and voice, it is important to consider the specific QoS requirements defined for each application.

In this context, following the layered model defined on the ETSI-BSM-QoS architecture (see Fig. 2.3), the satellite SD layers are responsible for dealing with the link resource utilization while having an implicit knowledge of the wireless medium. Nonetheless, to provide QoS differentiation, it is necessary to perform functions at the SI upper layers that the access scheme can not fully exploit.

Here, the use of cross-layer design among layers of the ETSI-BSM-QoS protocol stack has been proposed [KK05] to provide awareness between layers while enabling the interchange of information among them, in order to specify functions inside a protocol or a single layer of the system architecture.

In this way, the state of the art research on QoS provisioning is quite broad as it addresses many different topics. However, for the comprehension of the here propose PhD thesis, the state of the art research is divided in two main topics:

The former describes a background review on the ETSI-BSM-QoS architecture. This is done, in order to provide the baseline review including overall approaches to provide Quality of Service (QoS) architectures in emerging and future IP-based networks. Secondly, given that the main contributions proposed in this thesis are based on cross-layer techniques, *the QoS provisioning based on cross-layer design* is researched.

In this chapter, the most important related works based on cross-layer mechanisms are described. These provide optimization methodologies aiming to enhance specific characteristics at specific layers, such as the physical, the link, the network and the transport layers. As it will be observed most of the related works presented in this section are focused on satellite systems with fixed capacity. However, only few works to provide QoS guarantees over BSM satellite systems have been developed specifically focused on this area.

2.1 Background in QoS architectures

Nowadays, overall approaches to provide Quality of Service (QoS) architectures in emerging and future IP-based networks have been described by different standardized organizations such as IETF [Hus00], 3GPP [3GP99], ETSI TISPAN (Next Generation Networks NGN) [18505], ITU-T [IT04] and the Asymmetric Digital Subscriber Line (ADSL) Forum [For03]. One of the main common characteristic of these approaches, is the uncoupled provisioning of services and networks allowing them to be offered separated and to evolve independently. Here, a separation between functions developed for services and functions developed for transport are proposed, in which the provisioning between them must be done using open interfaces in order to support existing or new services independent of the network and access technologies used.

In this way, the ETSI TISPAN Next Generation Networks (NGN) has defined a basic approach to differentiate between the Application (services) and Transport strata, as it is shown on Fig. 2.1.

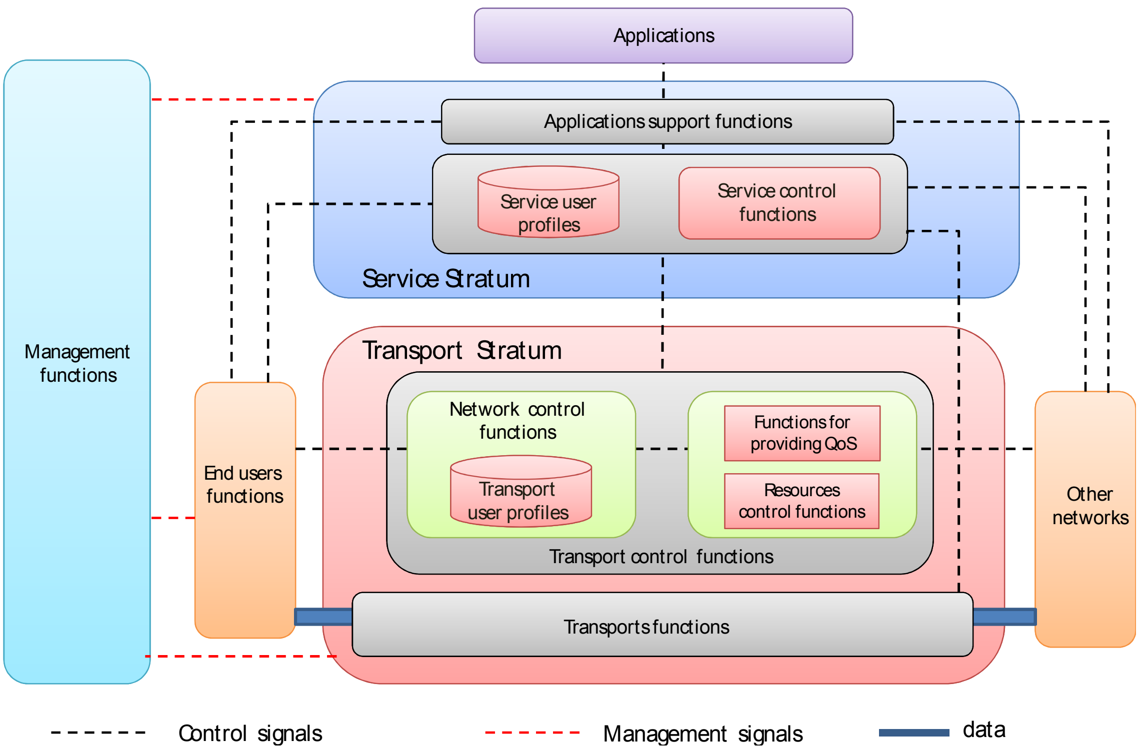


Figure 2.1: NGN application and transport strata

This is defined to overcome the increased emphasis posed by service providers which want to enable their clients for developing their own service customization, allowing them to support the creation, provisioning and management of services by means of service related Application Program Interfaces (APIs). In such networks the functional entities controlling policy, session, media, resources, service delivery, security, etc, are distributed over the operator infrastructure.

As it is shown on Fig. 2.1, the Application stratum is responsible for providing specific services to users, which has been requested by either users or call signaling protocols such as H.225+, H.245, Session Initiation Protocol (SIP), and H.248.

The Transport stratum provides a packet-oriented transport service and the desired network Quality of Service (QoS). The QoS is requested by signaling protocols such as Resource Reservation Protocol (RSVP), Common Open Policy Service (COPS), Simple Network Management Protocol (SNMP) and Next Steps in Signaling-Signaling Layer Protocol (NSIS-NSLP). QoS signaling can be exchanged with users end-points and the Application layer.

To enforce this, the ITU-T in Y.1291 [IT04] has defined the functional architecture framework to provide QoS guarantees based on building blocks or functions. The main functional blocks of this architecture are shown on Fig. 2.2.

As it is observed, these functions are classified in three functional planes: the Management plane (M-plane), Control (C-plane) and User plane (U-plane), which can be combined in different ways to reach different QoS objectives while satisfying users demands.

In the U-plane, functions to access individual packets for marking, classification, scheduling, policing, shaping and drop are defined. In the C-plane, functions such as QoS routing, service invocation, resource reservation and admission control are defined in order to answer the users demands. Finally in the M-plane, functions such as service subscription, policy, Service Level Agreement, provisioning and billing are defined.

Following these general approaches, the Broadband Satellite Multimedia (BSM) working group has defined a QoS architecture, standardized in the ETSI QoS BSM Services and Architectures standard (ETSI-BSM-QoS) [10205], which establishes the most widely accepted criterion for specifying the QoS requirements for BSM services based on the Internet Protocol (IP) suite.

2.1.1 The ETSI-BSM-QoS framework

The ETSI-BSM-QoS framework considers the BSM layer protocol stack to specify the QoS requirements for broadband satellite services. This layer protocol architecture is shown on Fig. 1.2. As it is shown, it follows the traditional OSI reference model. Nevertheless, the ETSI-BSM architecture is characterized by the separation between higher layers or Satellite-Independent (SI) layers and lower layers or Satellite-Dependent (SD) layers [ETS05b]. This modular reference architecture empowers control functions performed by the SI layers which can be either modified or updated regardless of the SD layer technology.

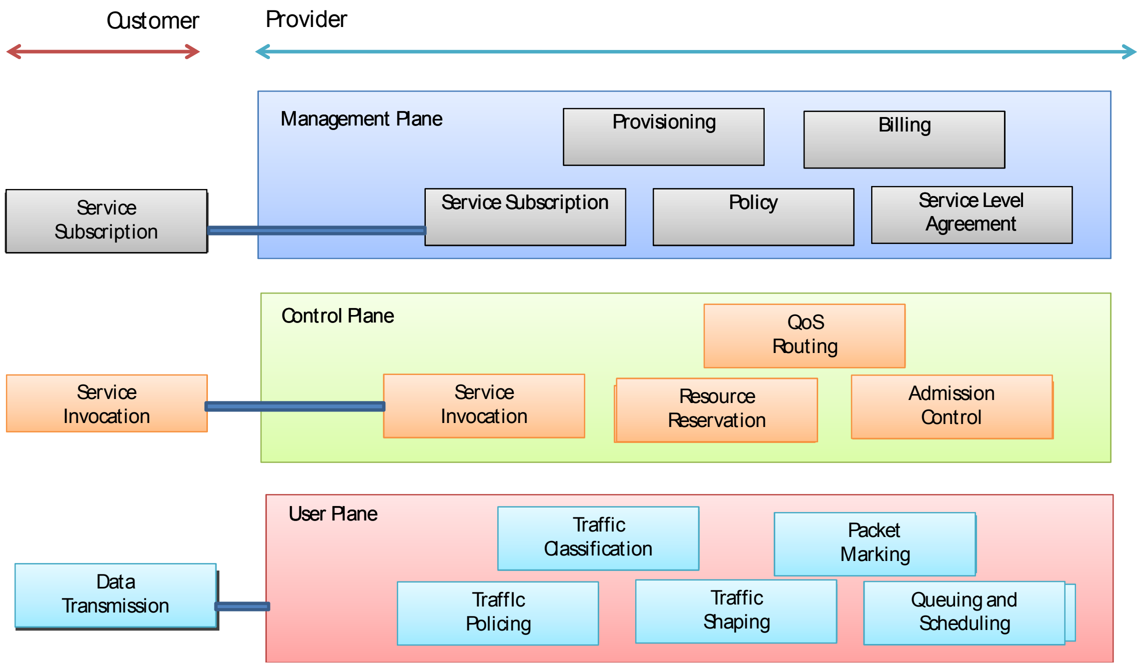


Figure 2.2: Functional architecture framework for QoS provisioning

In particular, the layers whose features depend on physical technology are called SD layers, which are associated with the physical and link layers of the OSI reference model. On the other hand, layers whose characteristics are independent of the physical technology are called SI layers, which are associated with the network and upper layer.

The interface between them is called Satellite Independent-Service Access Point (SI-SAP). This interface plays a crucial role for QoS performance monitoring, as this task is defined and measured at this interface.

The detailed QoS functional model implemented over the ETSI-BSM-QoS protocol stack is shown on Fig. 2.3.

As it is observed some functional blocks from U-plane, C-plane and M-plane are defined at each layer to enhance the provisioning of QoS guarantees. These functions operate above and below the SI-SAP interface.

In a BSM environment, the U-plane supports functions such as packet classification and conditioning, admission and subscription control, and forwarding mechanism based on the SLA. The C-plane allocates functions for signaling, service managing and control including the bandwidth allocation and queue management. Finally, in the M-plane functions such as admission, congestion and flow control parameters, packet sampling and statistics are gathered.

Particularly, focusing on the SI layers, the ETSI-BSM-QoS architecture separates the QoS provisioning into two categories: Guaranteed QoS and relative QoS.

The former has a per-flow scope that considers numerical bounds of the QoS parameters to provide specific QoS guarantees. On the other hand, Relative QoS has a per aggregated

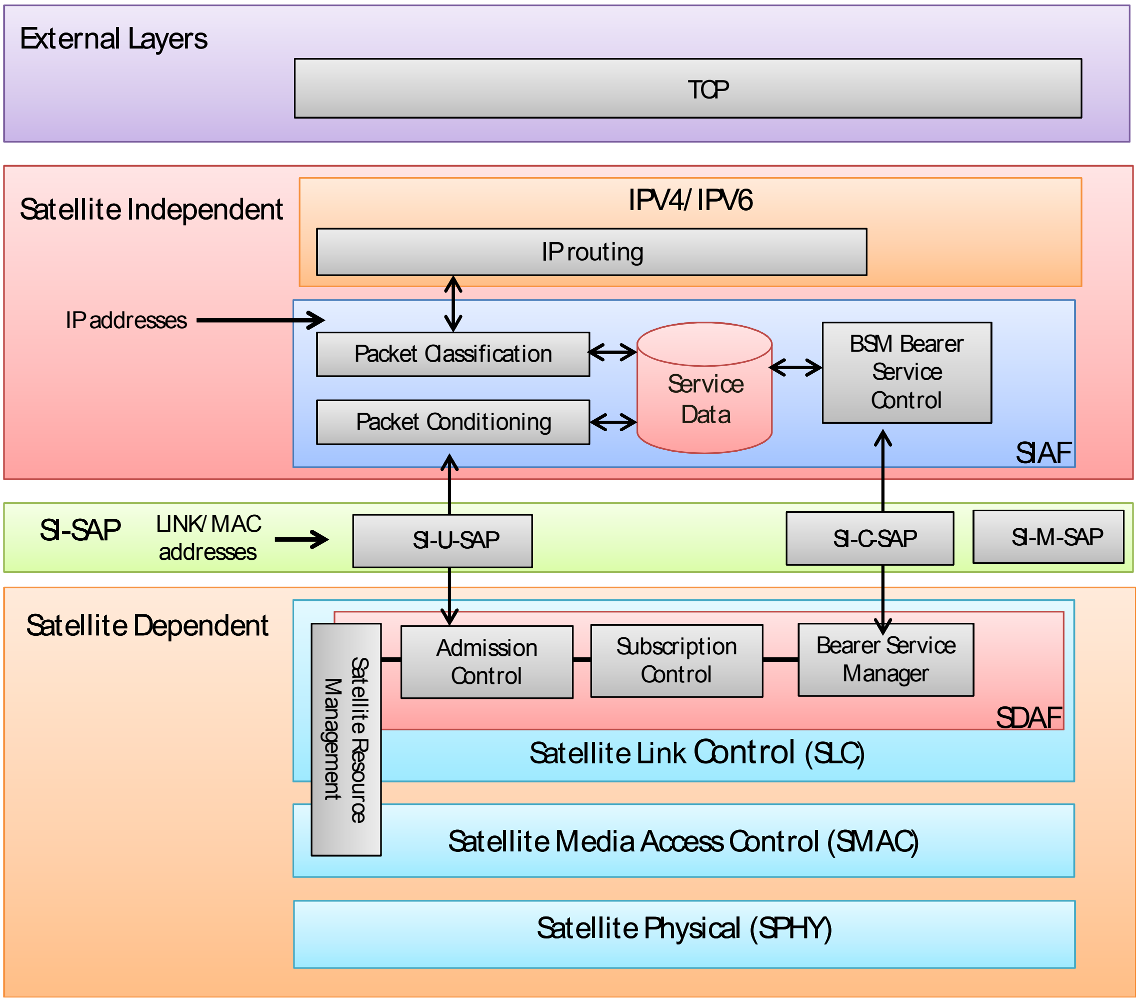


Figure 2.3: ETSI-BSM-QoS functional model

scope to provide aggregated QoS guarantees by translating the network complexity to edge nodes while enabling to maintain the scalability and simplicity of the IP network. These models can be related to the IETF framework such as the Integrated Services (IntServ) and Differentiated Services (DiffServ) respectively.

The DiffServ framework

The DiffServ architecture defined by the IETF [BBC+98] allows IP traffic to be classified into a finite number of classes differentiated by priority, to support different QoS levels. The main components defined in the DiffServ architecture are: *the traffic classifiers*, which select packets and assign (if necessary) their Differentiated Services Code Point (DSCP) values; *the traffic conditioners* which mark and enforce the rate limitation policy; and *the Per Hop Behaviour (PHB)* that enforces the differentiated packet treatments. In this sense, there are three predefined PHBs: Expedited Forwarding (EF), Assured Forwarding

(AF) and Best-Effort (BE). One of the main benefits of adopting the DiffServ framework is that the network complexity is translated to edge nodes, enabling to maintain the scalability and simplicity of the IP network.

The IntServ framework

On the other hand, IntServ specified by the IETF in RFC 1633 [BCS94], was motivated by the needs of real-time applications such as remote video, multimedia conferencing, visualization, and virtual reality. It provides a way to deliver the end-to-end Quality of Service (QoS) that real-time applications require by explicitly managing network resources to provide QoS to specific user packet streams (flows). It uses *resource reservation* and *admission control* mechanisms as the key building blocks to establish and maintain QoS.

IntServ uses Resource Reservation Protocol (RSVP) to explicitly signal the QoS needs of an application's traffic along the devices in the end-to-end path through the network. The idea of IntServ is that every router in the system implements IntServ, and every application that requires some kind of guarantees has to make an individual reservation. Flow specifications describe what the reservation is for, while RSVP is the underlying mechanism to signal it across the network.

2.2 State of the art

The design of cross-layer mechanisms over BSM satellite systems face several challenges, which are focused on minimizing the utilization of the radio resources and optimize the overall end-to-end application performance [Gia07]. This optimization requires changes into the design of protocols and systems while adding signaling mechanisms between protocol layers.

Authors in [KK05] summarize the alternatives to implement a cross-layer design. Here, it is important to mention, that in some cases the cross-layer optimization could increase the system performance while in others the performance levels could be degraded below those results obtained with layer approaches. Hence, the interactions created by using cross-layer design are in some cases intended and in others unintended. Such dependency relations require to be examined in detail, and timescale separated to enforce the optimization. In addition, the consequences of all such interactions need to be well understood, and theorems establishing stability may be needed. Therefore, it is possible to conclude that the proposers of cross-layer design should consider the whole design, including the interactions with other layers and the long-term architectural value that operators specifies.

Among cross-layer methodologies for satellite systems aiming to provide QoS provisioning across layers, it is possible to identify two approaches:

The cross-layer optimization based on the MAC resource utilization to support IP QoS (called *MAC-centric approach*) [CG07]. This approach enables to manage the available resources (layer two) for the support of QoS guarantees. On the other hand, the cross-layer design based on a resource sharing mechanism at the transport layer is defined to enhance QoS provisioning defined at the application layer (called *Application-centric approach*) [GWE+09]. Nevertheless, a cross-layer optimization based on the SI layer (called *SI-centric approach*), to provide priorities for QoS guarantees, while enhancing the optimization of the network resources has not been properly developed.

Therefore, in this Section, the most important related works based on cross-layer mechanisms are described. These works provide optimization methodologies aiming to enhance specific characteristics at specific layers, such as the physical, the link, the network and the transport layers. Most of the related works presented in this section are focused on satellite systems with fixed capacity. However, only few works to provide QoS guarantees over BSM satellite systems have been developed specifically on this area.

2.2.1 QoS provisioning for satellite systems

The new challenges posed by the support of QoS over DVB-S2/RCS satellite systems, have been addressed by some authors. However, there are few works that are focused on *developing and testing complete architectures to provide E2E QoS guarantees over BSM systems.*

The QoS provisioning over satellite systems has been concentrated on evaluating the adoption of the DiffServ architecture. For instance in [DKG+01], a QoS framework for GEO satellite networks is presented. Here, the objective is to analyze the impact on the performance of the assured forwarding traffic class considering different QoS factors such as traffic aggregation and multiple drop precedence levels. Similarly, the work developed in [RPDRF03] presents a gateway architecture to support DiffServ over satellite networks. In this architecture a resource management algorithm and marking mechanisms are proposed to guarantee the QoS level requirements.

Moreover, in [AM04], the authors presented the simulation results of testing a military satellite platform considering different QoS metrics (i.e. throughput, delay and packet losses). Here, the authors proposed a Weighted Random Early Discard (WRED) configuration to enhance service differentiation in a E2E satellite scenario. The results suggest that the implementation of a tuning WRED is essential to improve the throughput for high priority traffic while keeping packet loss rates at lower levels. Even though these works employ the DiffServ architecture to provide QoS guarantees, they do not consider the changes introduced by the ACM scheme over the DVB-S2 link.

Taking a different perspective, there are other works focused on providing *QoS guarantees based on a cross-layer optimization*. Particularly, an architecture design based on the Proportional Differentiated Service (PDS) model has been addressed by the authors in [CKP09] in the context of a Bandwidth on Demand (BoD) satellite scenario. Here, the provisioning of a proportional class-based service differentiation to TCP flows considering split-TCP connections is proposed. The scheduling algorithm for controlling the resource allocation is set up at the MAC layer (SD layers) and the PDS model is defined at TCP layer based on the throughput for each TCP flow. This proposed scheduler is based on the Waiting Time Priority Scheduler (WTP) scheduler [DSR02] which considers the average queuing delay to provide service differentiation. In addition, it proposes the use of Performance Enhanced Proxies (PEPs) to configure several parameters for different TCP connections.

Taking a different approach, in [AVCV12] a full design for QoS provisioning over DVB-S2 satellite systems is proposed. The design is developed on the SD layers. It includes a packet scheduler set up at the MAC layer [CG07] that considers the physical behaviour of the Ka satellite propagation channel to provide QoS guarantees. Here, the scheduler determines the fraction of time assigned for every transmission by each physical layer according to its correlated area where users undergo similar channel conditions. In addition, the scheduler design considers the tunable-fairness policy proposed in [FVG06] to provide fairness among *ModCods* at the SD layers. Here, the fairness levels achieved among users under different channel conditions are tunable to provide control over the system throughput.

Moreover, in [CABG05] and [PIA+05], the authors present an architecture for providing QoS over the return channel based on the standard DVB-RCS. This architecture was developed under the project called Satellite Broadband Multimedia Systems for IPV6 (SATIP6). Both papers present the QoS architecture for the return channel and provide validations through an emulation platform. As the proposed architecture is based on the well adopted standards and the best practices solutions, it should be considered as a reference model for QoS provisioning over fixed capacity systems. However, as the proposed architecture was developed for DVB-S/RCS systems with constant capacity, considerations such as a cross-layer design or the adaptability of the physical layer were not addressed.

In [SSF08] the authors discuss the basic requirements to develop a RCST gateway supporting VoIP traffic considering tight QoS provisioning. The system design includes a QoS management which is not based on reservation. It is used to suitable accomplish changes in traffic scenarios and to support new applications. The main elements of this architecture are illustrated throughout the paper following a layer structure. A preliminary set of simulations is presented to point out benefits and to identify major issues. Implications of buffer management, different scheduling algorithms, and bandwidth allocation protocols are analyzed.

2.2.2 Cross-layer design for satellite systems

In this subsection, the QoS provisioning is analized from a multilayer perspective considering cross-layer techniques. Therefore, it is divided into the two main layers of the ETSI-BSM protocol stack: the Satellite Dependent (SD) and the Satellite Independent (SI) layers.

Literature review focused on the SD layers

The paper presented in [LC04] studies the time-slot assignment problem over a DVB-RCS satellite system. Here, the authors focus their research work specifically on the resource allocation for the return link considering a limited physical layer with two possible rates. In addition, in this paper a complete mathematical formulation is proposed which is resolved by considering the problem as a nonlinear integer programing problem. Given the computational complexity to find the optimal solution for such a problem, the authors propose an heuristic algorithm which is executed in real time to find a nearly-optimal user scheduling. The scheduler assigns users considering a time-frequency grid that correspond to the Terminal Burst Time Plan (TBTP) defined in the DVB-RCS standard. Each user's contribution to the penalty function is equal to the number of slots that are requested but no reserved, multiplied by the weight assignment to the QoS traffic class. The proposed algorithm is designed to minimize the penalty values as a method for guaranteeing fairness and QoS.

Similarly, in [SDC$^+$03] the authors take the control theory approach to analyze RED and propose guidelines for tunning RED algorithm specifically developed for satellite networks. In contrast to RED proposals for optimizing AQM mechanism, in [cFSKS02] the authors propose an AQM mechanism which is called BLUE and it is based in different congestion detection algorithms. Specifically, BLUE uses packet loss events and link idle events to manage congestion. Using both simulation and controlled experiments, BLUE is shown to perform significantly better than RED, both in terms of packet loss rates and buffer size requirements in the network. The authors first address the problem with RED and demonstrated that in some situations RED can perform poorly and it is not able to limit packet losses even with ECN. The authors show by means of simulation that BLUE can exhibit higher stability, better link utilization and lower packet losses compared with the same scenario using RED. It is also demonstrated that it is necessary to have smaller buffers than those buffers with RED. However, as commercial solutions are able to be deployed using RED, alternatives such as BLUE lack of validation test in real satellite environments.

Authors of [FL06] propose to monitor the buffer occupancy level of the satellite link MAC queues. They adopt Random Early Discard (RED) as congestion detection algorithm. However, instead of using RED for marking/dropping packets, they use a modified version of RED at the MAC layer queues to generate two congestion notification signals: I (equivalent to a marking threshold) and II (equivalent to a dropping threshold). These notifications are sent by the MAC to the transport sender at the PEP. When a transport sender receives an ACK, it will look the congestion notification signal. In the presence of congestion signal I or II, the source cuts its congestion window by 1/4 or 1/2 respectively. Otherwise, the source allows its window to grow as in the original TCP mechanism.

In [BSAD04] a comprehensive analysis on AQM mechanism is presented. Here, the AQM performance is evaluated in different scenarios as well as their benefits over short-lived connections. The authors propose to use the control theory approach and the definition of mathematical models to describe and analyze different AQM approaches. They focus their research on issues such as congestion detection, stability, fairness and scalability. The paper also shows that some assumptions made about AQM mechanism are not always true.

The same authors have developed in [LZG05a] a cross-layer design for multiuser scheduling at the data link layer in a scenario where each user takes advantage of the ACM technique at physical layer. Specifically, this paper proposes a simple scheduler where users are classified by considering their QoS requirements. The layer two scheduler consider two services classes to provides efficient bandwidth utilization and isolation of high quality users from users of the lower class. The scheduler objective is to share the number of available time slots among the users giving to each of them the minimum resource allocation that accomplished their QoS requirements. The unused slots are finally allocated among the users that have requested best effort services.

Finally, in [LZG05b] the authors propose a cross-layer design for wireless links to overcome the limitations of the traditional design of the physical layer. It assumes that there is always sufficient data waiting to be transmitted. This paper analyzes the effect of a finite-length queuing design at the data link layer and the ACM techniques to derive expressions for the packet loss rate, the average throughput and the average spectral efficiency.

Literature review focused on the SI layers

The following group of related works has been considered to have partial knowledge (PK) about the network state, which are characterized for using cross-layer techniques. These proposals measure either the available bandwidth or buffer occupancy parameters but not both parameters at the same time. Again, none of these proposals consider an integrated design which addresses end-to-end QoS scenario like ETSI-BSM.

The work presented in [TKN06] proposed a recursive, explicit and fair congestion control method (REFWA). This related work is proposed in a E2E framework for LEO satellite networks. This method makes an estimation of both parameters: the number of TCP flows and the Bandwidth-Delay Product (BDP) value of the network. Taking into account the number of TCP flows, the system is automatically adapted to them. The scheme matches the sum of windows sizes of all active TCP connection (sharing the bottleneck link) to the BDP value in order to control the network utilization. The proposal also adapts its requirements to the free buffer size. In addition a computational method is proposed to control the system efficiency by matching the aggregated traffic rate to the link capacity and the total buffer size. The main drawback of this proposal is that as the RCST modifies the *receivers advertised window* (*rwnd*) *field* for all the TCP segments in every packet, as a result high computational cost is demanded.

On the other hand there are several approaches that are based on the implementation of PEPs, such as the Performance Enhancing Transport Architecture (PETRA) [MM04], which was developed under the ARTES project of the ESA. PETRA proposes a transport protocol for the satellite segment named Satellite Transport Protocol Plus (STPP) to counteract transmission errors. STPP uses a sliding window together with a selective repeat retransmission mechanism that is based on "Negative ACKs" (NACKs).

Another similar approach is the Satellite PEP (SaTPEP) [VKM02] architecture. SaTPEP proposes a splitting architecture in which the congestion control of the satellite TCP is based on the estimation of the satellite link utilization. The congestion control mechanism of SaTPEP tries to maintain the satellite link fully exploited by setting the congestion window of TCP senders to the link Bandwidth-Delay Product (BDP). As a result, SaTPEP considers that TCP segment drops are only caused by transmission errors.

In [VF08], XTCP (Explicit window-based control in lossy packet networks) is presented. XTCP behaves like TCP, except for the fact that it uses an explicit congestion signal that goes from the routers to the TCP senders. In this case, the sender only decreases its window when it receives the explicit congestion signal.

Similarly, [Mar03] the concept of complete knowledge (CK) was introduced for the first time. Nevertheless, this work might be classified within the group of PK proposals, since it only considers as cross-layer information the current available bandwidth. In particular, this pioneer work proposed a modification of TCP as a basis of a new satellite transport protocol. The standard slow start phase was removed and the initial window was set to BDP. The congestion window value was no longer increased when acknowledgments arrived but it had a fixed value all the time to maintain a high level of link utilization.

Essentially, this approach is equivalent to the SaTPEP approach [VKM02], however it uses a cross-layer design to obtain the available bandwidth instead of using an estimation mechanism computed by the TCP receiver.

Similarly, in [RLZ09] the authors proposed a burst-based TCP called Noordwijk TCP which is based on the I-PEP specification especially developed for (Dynamic Assignment Multiple Access) DAMA DVB-RCS systems. This new TCP variant is characterized for enhancing the HTTP transfer performance by introducing sender-only modifications, supporting the Space Communications Protocol-Transport Protocol (SCSP-TP) defined in the I-PEP specification. In the design of Noordwijk TCP the window-based transmission is replaced by a burst-based transmission which uses two parameters to compute its values: the burst size and the burst transmission interval. These parameters are accurately selected, every time a transmission starts and it is updated according to ACK-based measurements. To regulate variation on the burst rate, two algorithms are proposed: *Rate Tracking* and *Rate Adjustment*. Using the TCP Noordwijk as a transport protocol in DAMA DVB-RCS satellite systems, it is possible the device integration, having both functionalities (the RCST and the PEP) in the same device. One of the main drawback of this proposal is that the design does not consider the delay variability introduced by using DAMA. In addition the queue management is not taking into account.

Finally, there are other related works which are called the Complete Knowledge (CK) proposals. In this case, recent proposals for explicit congestion control devised for the new high-speed Internet paradigm are: JetMAX [ZLL08], MaxNet [WZ02], RCP [DKZsM05], XCP [HR02] and EFXCP [WRMG09]. These next-generation protocol examples operate in an end-to-end basis which are designed not considering either PEPs or scenarios prone to transmission errors. In addition they address the challenge of controlling the transmission in high-speed networks which will enable the development of the future Internet. In essence, these approaches measure simultaneously the link capacity and the buffer occupancy to compute a single feedback value per flow. Since this family of protocols is conceived for generic networks, its operation involves senders, receivers, and the intermediate routers. Intermediate routers have a particularly active role, measuring and computing explicit congestion control feedback. The feedback is transmitted in-band, that is to say, using fields in the headers of the downstream packets. Then, this feedback is copied in the upstream packets, and finally, after a round trip time the congestion control information reaches the sender. In general, the explicit congestion control algorithms of these protocols have the goal of keeping empty the output queues of the routers. In this way, these algorithms minimize the delay while roughly using the total link capacity. However, a fine control to achieve a full-exploitation at any time is not an issue for these proposal, since their application scenario are developed for specific networks such as high speed networks that consider links with constant bandwidth capacity. The QoS provisioning for future high speed Internet conceives congestion control protocols operating in open environments but not with end-to-end QoS architectures as ETSI-BSM. In this high speed Internet scenario, to obtain some degree of QoS, flows are prioritized by intermediate routers using some bandwidth distribution algorithms. On the contrary, as in the DVB-S2/ETSI-BSM QoS scenario, bandwidth is a scarce resource which varies quite drastically over time. As a conclusion, while CK approaches that can be found in the literature are promising solutions for the future high speed Internet, they are not suitable for working over DVB-S2/ETSI-BSM QoS satellite systems.

Literature review focused on the interaction between SI and SD layers

The work proposed in [ZRK06] is an example of TCP enhancements based in link-layer techniques. In this paper the authors focus their work on the issue of performance modeling in terrestrial satellite hybrid networks. They propose an scheme based on the link layer over an E2E path that enables the TCP to have the information related to packet losses. It works based on the Selective Repeat ARQ (SR-ARQ) mechanism at the link layer, considering the propagation delay value present at the satellite segment. This related work also develops a mathematical model in which a theoretical comparison in terms of throughput is presented. The comparison is done considering the proposed E2E architecture with the link layer SR-ARQ support, against a TCP splitting architecture. Although, using the proposed architecture with SR-ARQ, the results show that the system efficiency is improved, the results using the splitting architecture provides superior improvements in terms of throughput. Outcomes of this study show the advantages of using TCP splitting in an environment with long round trip time as satellite environments.

In [GH09], the authors propose an architecture based on PEPs for DVB-RCS satellite networks. This proposal is developed considering a top-down cross-layer optimization, in which the TCP sends the desired congestion window value to the DVB-RCS terminal. Then, the terminal sends a resource assignment request to obtain the capacity assignment which is shared between the competing flows. The authors also propose a in-band signaling mechanism to change the advertised receiver window value. This is done by modifying the advertised receiver window field of reverse ACK packets.

On the other hand, the work [KAE$^+$07] presents a button-up cross-layer optimization mechanism over a DVB-RCS system using PEPs. The proposal considers an algorithm to keep control the queue level. Here, the TCP congestion window value is directly mapped to the available length of the corresponding MAC queue allocated at the PEP. Therefore, every time the TCP congestion window value needs to be updated, it is done considering the current available length of the MAC layer queue. The algorithm also includes a threshold parameter to guarantee that the packets do not remain in the queue more than one RTO (Retransmission Time Out). This proposal can be classified in the PK approach due to the lack of knowledge about the service rate information.

In [LZM07], the proposed cross-layer interaction also employs the buffer occupancy level of the satellite link MAC queues. This approach tries to prevent congestion by adjusting the TCP windows of senders according to the occupancy changes experienced at the queues. The TCP windows, which are called *xlayer windows*, are computed periodically at fixed time intervals. This proposal considers two cross-layer mechanisms developed for a DVB-RCS system with PEPs. Firstly, the introduction of a cross-layer algorithm defined to optimize the DAMA mechanism. It is done by adopting the Adaptive Coding (AC) cross-layer information (reported by the return channel) every time the TBTP is computed at the NCC. By taking into account these AC rates, the time slots are grouped at the NCC. The AC cross-layer information is signaled using the *Channel ID* field of Sack messages. Secondly, a MAC-Transport cross-layer interaction is proposed to counteract the capacity variations due to the DAMA assignment as well as the presence of a rain fading event. The proposed cross-layer mechanism forces the TCP sending window to match a given value calculated when changes in the sender queue size are experienced. In this case, the MAC layer triggers these cross-layer messages to the transport layer to optimize TCP data transfers. One of the main drawback of this proposal is it is unable to identify the reason why the satellite capacity is increased. As this increment could be generated because either the queue gets overloaded or the system capacity is really increasing. As previously discussed, most traditional self adaptive scheme based their feedback on only one parameter either service rate or buffer occupancy, but do not consider the link delay or link bandwidth variations.

Similarly, in [Yeh02] and [BY04], the authors analyze a cross-layer resource allocation for wireless systems. It includes the interaction between the three lower layers: the network, access and physical layer to maximize the throughput and minimize the delay in multi-access wireless networks. Those papers can be seen as a complementary work since both of them go beyond the access layer and extends its analysis to the network layer, presenting a complete cross-layer perspective. Both works specify the way to achieve an optimal theoretical throughput and delay values. First the authors obtain the optimal throughput value by means of the combination of power control and rate allocation (ACM allocation) using a max-weight Longer-Queue-Higher-Rate (LWQHPR) algorithm. Secondly, they show that LQPHR algorithm is only delay optimal. The analysis shows that it is possible to minimize the average delay by implementing LQHPR policy according to the capacity region which is time dependent.

In [DSLG01] and [DSC+05] the same authors propose a multi-level Explicit Congestion Notification (ECN) algorithm. The ECN algorithm includes a packet marking procedure that allows the intermediated nodes to notify the sender the reached congestion level without having to drop any packet. Specifically, in [DSLG01] the authors demonstrated that it is possible to notify four different congestion levels by reusing the ECN bits on the IP packets. It is also shown through simulation that the QoS parameters such as the throughput, link utilization, delay and losses are improved in comparison with traditional ECN schemes. This approach was considered especially useful for satellite links that show low responsiveness to congestion signs due to the high RTT value. Similarly, In [DSC+05] the authors analyze the multi-level ECN implementation using control theory and considering the stability point of view. The authors proposes guidelines to correctly tuning multi-ECN for satellite networks.

Finally, the same authors in [DSLJ01] propose an improvement of ECN specifically developed for satellite systems. It is done by means of by adding a mark-front policy to the ECN signal. The main advantage of mark-front strategy is that it sends faster feedback information about the congestion and consequently enables faster reaction from the source. This means that when the congestion avoidance mechanism is activated, the packets living the queue are marked instead of the arriving packets.

Based on the state of the art research, in this PhD Thesis several contributions to provide QoS guarantees over the DVB-S2/RCS satellite system are presented. In contrast to the previous related works, the proposed design is focused on the SI layers in which the PEP element is optional, aiming to preserve the E2E path. The design proposes a complete *gateway* architecture including an IP packet scheduler that considers the bandwidth availability present in the satellite system to enhance QoS guarantees. The architecture is based on a cross-layer optimization between the physical layer and the network layer. Particularly, the proposed model studies the impact on the traffic class performance when bandwidth variations due to the changes introduced by the ACM scheme over the DVB-S2 link are experienced. The proposed architecture is designed inside the DVB-S2 *gateway*, which is allocated in the terrestrial segment. As a result the proposed algorithm can be easily adopted by satellite operators.

CHAPTER 3

DVB-S2 QoSatArt architecture design

The DVB-S2 standard [ets04] defines as mandatory the use of the Adaptive Code and Modulation (ACM) [RVCM04] techniques, to attain Interactive Services. Such techniques reduce the available link bandwidth (transmission rate), if necessary, to achieve quasi-error free channel conditions for each individual user to provide them with the most suitable Modulation and Code (*ModCod*) value according to the measured Signal-to-noise-plus-interference-ratio (SNIR) value reported by the return channel.

The major benefit of adopting ACM techniques it is that the obtained spectral efficiency is optimized, being as high as possible for all the satellite terminals. Nevertheless, a fundamental change related to the satellite physical layer must be considered, this is that the satellite capacity will be constantly fluctuating due to the ACM adaptation mechanism. In this way, one of the main concerns using GEO satellite systems is the management of these bandwidth variations to satisfy the specified QoS levels for different traffic classes.

In this chapter, the first contribution of the thesis called QoS SATellite ARchiTecture (QoSatArt) is proposed. The design considers the fact that the bandwidth availability present in the satellite system is adapted using ACM techniques. Here, the complete architecture design is developed that considers the challenges posed by GEO satellite systems (i.e. delay, losses and bandwidth variations) while adopting several mechanisms based on a cross-layer design to enhance the provisioning of QoS guarantees over the forward satellite channel.

For this propose, the DiffServ framework [BBC+98] is adopted to guarantee the required QoS specifications defined in the Service Level Agreement (SLA). This framework is characterized for having three PHBs: Expedited Forwarding (EF), Assured Forwarding (AF) and Best-Effort (BE). One of the main benefits of adopting the DiffServ framework is that the network complexity is translated to edge nodes, enabling to maintain the scalability and simplicity of the IP network.

The QoSatArt architecture design is developed in compliance with the ETSI-BSM-QoS [10205] and the recent standard defined for the Digital Video Broadcasting-second generation (DVB-S2) [ets04] forward channel.

3.1 Architecture design objectives

The QoSatArt architecture has been designed considering the following objectives:

- The solution must be designed in compliance with the ETSI-BSM-QoS framework, providing a detailed design at the Satellite Independent (SI) layers in order to provide QoS guarantees by means of traffic priorities.
- The solution must be compatible with IP-based technologies and protocols and should specify the requirements for the forward and the return channel based on the DVB-S2/RCS standards.
- The architecture must consider that the satellite system is affected by rain events which reduce the bandwidth availability. Therefore, the adoption of algorithms and cross-layer techniques to guarantee the QoS requirements established in the SLA must be mandatory.
- The architecture must minimize the delay values experienced at the Application layer for delay sensitive traffic, while providing delay guarantees with the Active Queue Management (AQM) system.
- The architecture must provide low complexity on its implementation and can seamlessly inter-operate with terrestrial IP networks.

3.2 QoSatArt scenario

The E2E scenario defined for the QoSatArt architecture is shown on Fig. 3.1. It considers a broadband satellite system in the Ka band (30/20 GHz) working in a multi-beam architecture. The satellite scenario is considered transparent in star topology. We focus the design on a single Time Division Multiplexed (TDM) carrier.

Fig. 3.1 represents the typical scenario, in which remote users demand Internet applications by the intensive use of the forward channel. In particular, a military remote vehicle requires accessing critical applications and data allocated at the military operation center (MOC) to provide the valuable information during a military mission, where the satellite system is the only technology that remains available.

Here, three sources, with different QoS levels, send data to a remote destination by means of the broadcast GEO satellite channel. This channel represents the communication link between the ground *gateway* and return channel satellite terminal (RCST). Particularly, in the QoSatArt scenario, a heavy rain event is affecting the available bandwidth in the DVB-S2 channel.

In the proposed scenario, the supported applications are defined considering the three Per Hop Behaviour (PHB) (see Fig. 3.1) of the DiffServ traffic classes. The EF traffic class [DCB$^+$02] is designed to provide low-loss, low-latency, low-jitter and to assure bandwidth services such as VoIP applications, where packets normally find short or empty queues.

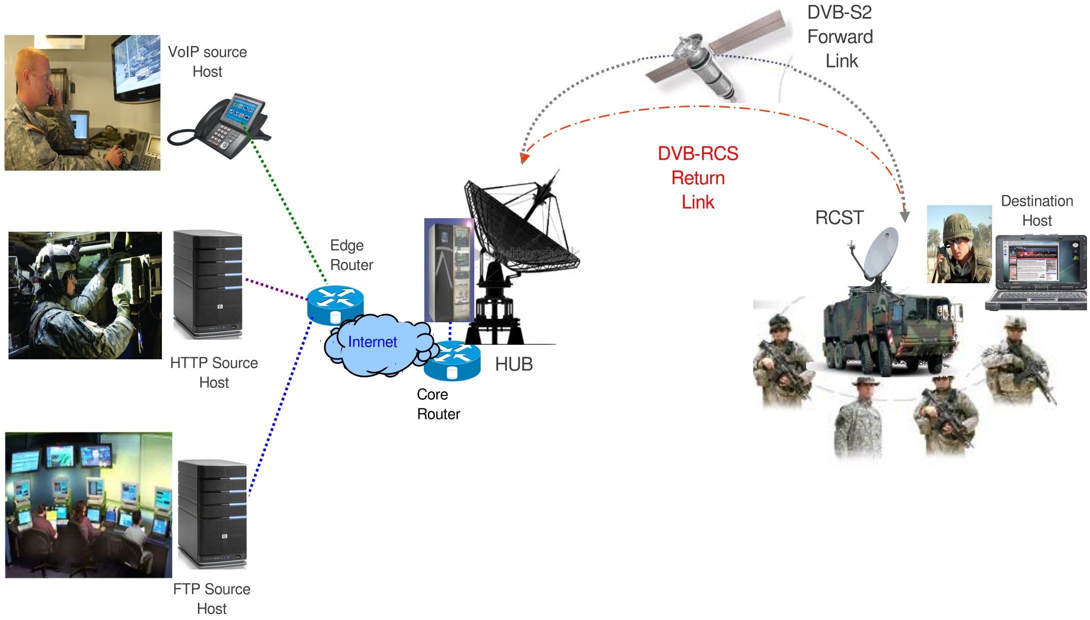


Figure 3.1: End-to-End BSM satellite scenario

The AF traffic class [HBWW99] is designed for non-real time traffic with QoS support, for instance the current Internet where Hypertext Transfer Protocol (HTTP) applications are demanded by end users. The AF traffic class separates the network traffic into four independently forwarded AF classes which are differentiated based on their Drop Precedence (DP). Finally, the BE traffic class [BBC$^+$98] is used for unclassified traffic such as FTP applications. In this case, the end users or in general any organization will have a non-guaranteed achievable throughput.

Here, the *QoS policy* defined by the satellite operator allows the EF traffic class to have the highest priority, while the AF traffic class has more priority than the BE traffic class. As a result, the BE traffic class uses the remaining link bandwidth, being able to also use the bandwidth that any other class does not use.

Initially, the architecture definition is focused on the DVB-S2 broadcast channel of the GEO satellite systems. However, the design of the QoSatArt architecture assumes a return link based on the DVB-RCS standard [ets05a] with the support of a Bandwidth-on-Demand (BoD) mechanism to share the radio spectrum among the allocated users. Here, the use of the Demand Assignment Multiple Access (DAMA) scheme to guarantee different QoS levels at the return link is also assumed [LRF09], [GPS06]. In addition, the network control center (NCC) is the element that monitors the link capacity for each RCST transmission.

3.3 DVB-S2 QoSatArt architecture design

The QoSatArt design is developed inside the DVB-S2 *gateway* which is the central element in the architecture. This is done in order to allow satellite operators to easily adopt the proposed architecture with low deployment cost. In addition, the proposed architecture allows the satellite operator to manage the functional parameters to establish priority levels and traffic rates according to the defined SLAs.

One of the main goals of the QoSatArt architecture is to guarantee different QoS levels for IP traffic over the DVB-S2 channel while reducing latency and jitter values, considering the fact that the available bandwidth present in the satellite system is constantly changing.

In addition, the design of QoSatArt considers the fact that the bandwidth availability present in the satellite system is adapted using ACM techniques. This adaptation is performed based on the intensity of rain events [CGR08], which is constantly updated using a cross-layer design between the physical layer and the network layer. This is done to provide QoS provisioning when different levels of link capacity are available in the satellite system.

3.3.1 QoSatArt Gateway-QoS functional blocks

The QoSatArt design includes several functional blocks and mechanism defined at the SI layer:

(i) A complete Active Queue Management *(AQM) system* that considers Token Buckets (TBs) as rate limiters to regulate and guarantee a minimum transmission rate for each traffic class according to the priority levels established by the satellite operator. Here, the queue design considers the Bandwidth Delay Product (BDP) value to dynamically set the queue lengths to enforce bounded delay values for high priority traffic classes.

(ii) A modified queuing policy called Re-Queuing Mechanism (RQM) to reduce delay and jitter while improving the user's QoE for the Expedited Forwarding (EF) and the Assured Forwarding (AF) traffic classes. This mechanism follows the philosophy of the DVB-S2 design, in which retransmissions are avoided because the two-way propagation delay is significantly high. In our case, the RQM mechanism prevents dropping packets that do not fulfill the DiffServ traffic class specification.

(iii) A *XL-Manager* module to manage and orchestrate the cross-layer interactions between SI and SD layers. Here, the first the cross-layer interaction is proposed between the physical layer and the network layer to guarantee the QoS requirements established in the SLA while considering the fact that the DVB-S2 forward channel is affected by the presence of rain events.

(iv) An *IP scheduler* to allocate bandwidth resources for prioritizing those flows with high QoS requirements. The IP scheduler uses an algorithm that adjusts its internal values considering the capacity present in the system. This dynamic adaptation considers the cross-layer information sent by the physical layer to provide enhanced priority for specific flows when a reduced and limited channel capacity is experienced in the satellite system.

3.4 Satellite Independent design

The design of the QoSatArt architecture is developed focus on the SI layers of the ETSI-BSM-QoS protocol stack. This is done, to enforce the establishment of priorities among users and applications while improving the management and control functions performed at the upper layers [M M08]. Moreover, focusing on the SI layers, it is possible to isolate the SD layers (i.e satellite physical, MAC and Link Control) to include different physical layer supports (for heterogeneous networks) [Mar07].

The proposed design inside *the gateway* including its separation between the high SI layers and the low SD layers is shown on Fig. 3.2. Particularly, the SI layers are defined to deal with QoS differentiation based on the DiffServ framework. Conversely, the SD layers are proposed for applying different DVB-S2 channel adaptations.

This design complies with the ETSI-BSM-QoS functional architecture supported by the standards ETSI TS 102 157 [ETS03] and ETSI TS 102 462 [10205].

The QoS blocks and their functionalities as a part of the proposed cross-layer QoSatArt architecture, are described as follows: At the SI layer, the DiffServ server, the AQM system including the set of DiffServ queues, the *XL-Manager* module and the IP scheduler are set up. At the SD layer, several buffers are defined for allocating packets waiting to be encapsulated and multiplexed before these are forwarded to the required RCST.

For simplicity at SI layers, the DiffServ queues have been defined considering the EF, AF and BE traffic classes. Similarly, at the SD layers several queues are set which are associated with different *ModCods* schemes. In this scenario, $ModCod_i$ is said to have higher spectral efficiency than $ModCod_j$ (where $i > j$). Here, the *ModCods* are ordered from high to low spectral efficiency.

In order to determine the QoS treatment to be applied through the satellite network, packets coming from other IP networks are marked. In most of the cases, packet marking is typically performed at the edge nodes of the DiffServ satellite domain. However, in some cases satellite operators may require a remarking process to be performed inside the *gateway* to adjust the forwarding policy that will be applied.

The *QoS manager* module, shown on Fig. 3.2, is defined to manage the QoS parameters for establishing priority levels and traffic rates according to the predefined SLAs. This allows the satellite operator to perform operation and management functions according to its requirements. In addition, the *QoS policy* is also set up at this module.

In particular, the *QoS policy* implemented in the QoSatArt architecture allows all in-profile packets (coming from the EF and AF traffic classes) to be sent directly to the IP

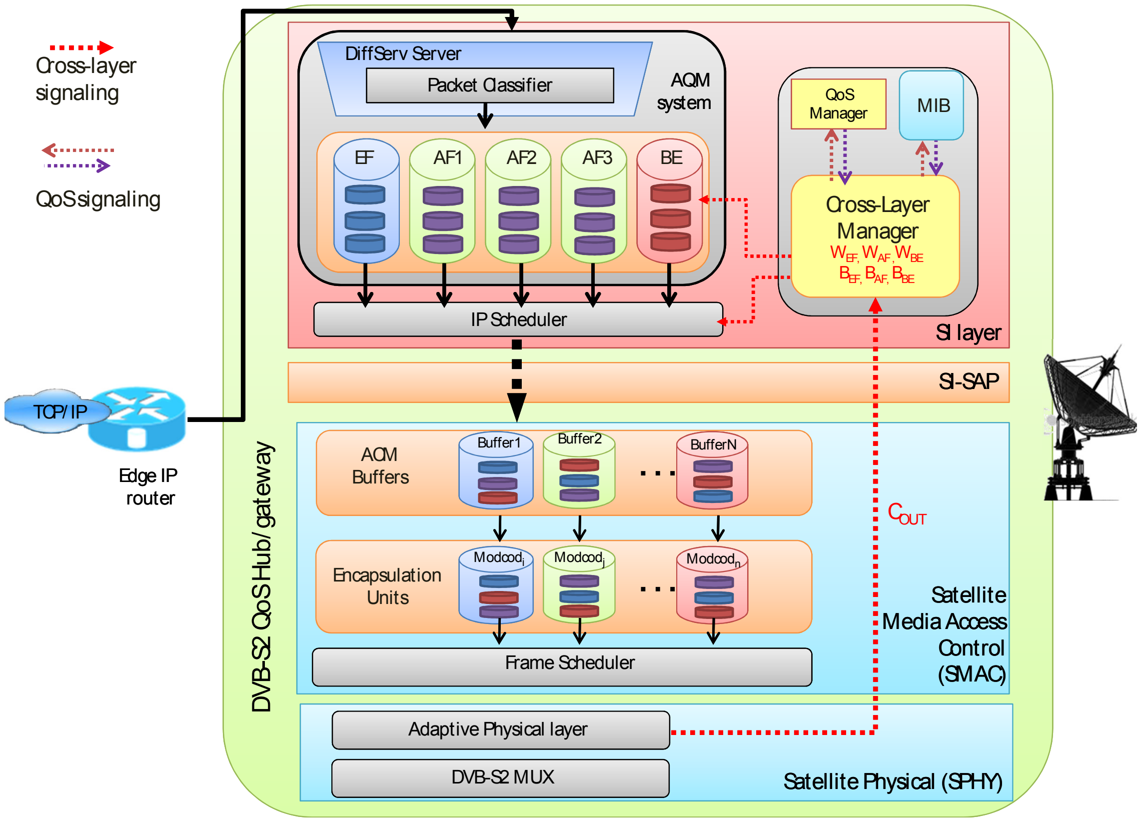


Figure 3.2: DVB-S2 QoSatArt gateway design

scheduler. This element is defined to control the order in which packets are extracted out from its queues. Here, it is important to bear in mind that the IP scheduler incoming rate is limited by each token bucket.

As it is shown on Fig. 3.2, packets entering the DVB-S2 QoSatArt *gateway* are received by the *DiffServ server*. This module is responsible of receiving different flows, classifying the incoming packets and deciding (if required) if a packet needs to be re-assigned with a different QoS level, by marking/re-marking its DSCP. In practice, the *gateway* implements packet classification and per hop forwarding scheduling according to the DSCP value of each packet. At this point, packets are forwarded to get in the queues of the Active Queue Management (AQM) system.

3.4.1 The Active Queue Management (AQM) system

The detailed design inside the *AQM system* is depicted on Fig. 3.3. Notice that the proposed scheme allows multiple flows to be aggregated and treated as a single flow per traffic class. For simplicity, the set of DiffServ queues have been reduced to three, representing the EF, AF and BE traffic classes. The queuing model includes the predefined DiffServ traffic classes, allowing each of them to have its own physical and separated queue. Packets coming from these queues are scheduled to the SD layers based on a the IP scheduler.

Here IP scheduler functions are linked intrinsically with the satellite bandwidth allocation carried out by lower layers.

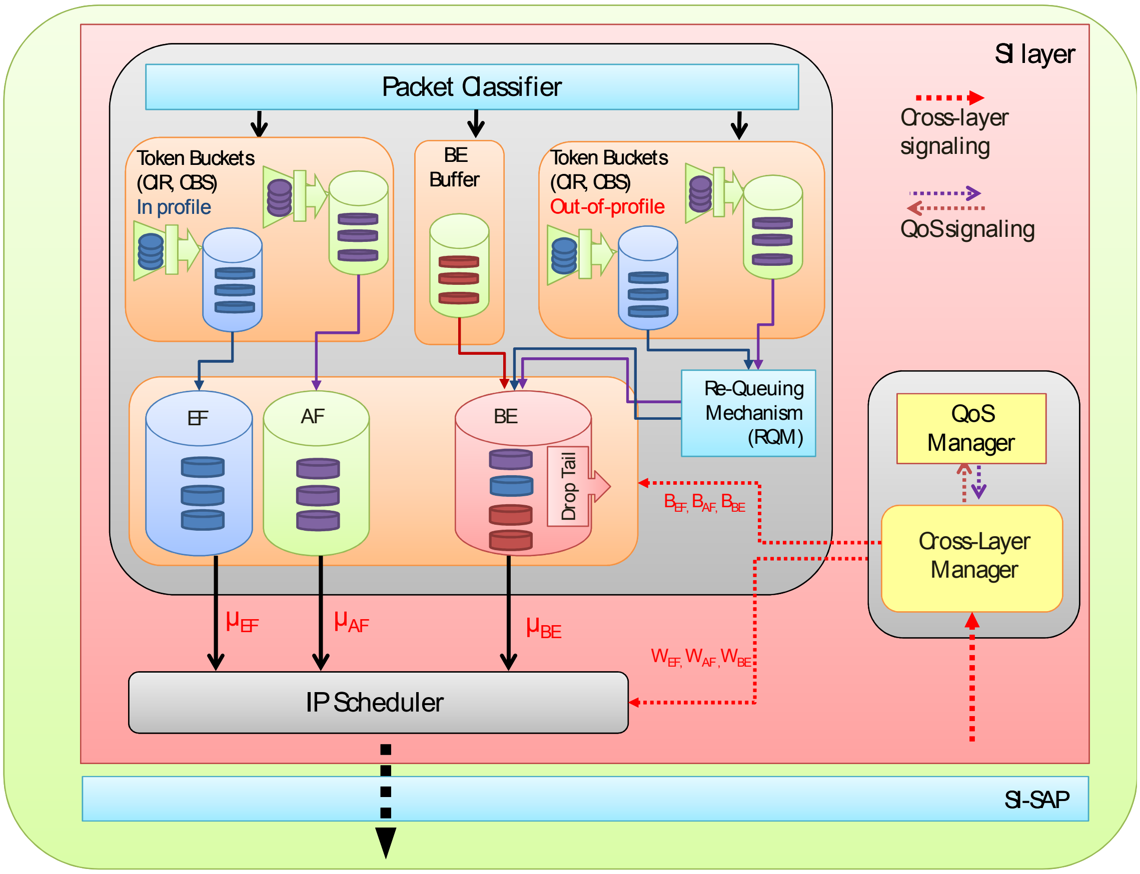


Figure 3.3: The AQM mechanism design inside the gateway

As it is observed on Fig. 3.3, the high priority traffic classes (EF and AF) implement token buckets (TBs) as rate limiters to guarantee certain transmission rates for each traffic class. These rates are defined in accordance with the bandwidth assignment established in the SLAs between the satellite operator and the subscribers. Importantly, the operator is able to modify these rules as a part of the operation and management tasks in the satellite gateway.

The incoming TBs limiting rates represent the transfer rates for the EF and AF traffic classes (μ_{EF} and μ_{AF}, respectively). Using TBs, packets are separated in two levels: in-profile (fulfill the SLA) and out-of-profile packets (do not fulfill the SLA).

Alternatively, the BE traffic class is not provided with a token bucket policer, this assumption is made considering that this traffic type does not need to adjust a committed rate. Therefore, when the BE queue is full, no particular algorithms are needed to decide which packet is going to be dropped. As a result, a simple drop tail (DT) mechanism is implemented.

On Fig. 3.3 the token buckets with parameters (σ, ρ) are defined as a traffic regulator for the high priority traffic class (EF and AF). Here, ρ represents the traffic rate for each traffic type and σ represents the bucket depth (in packets). In this way, packets move from each packet buffer one by one into the queue. A token counter controls when packets can move into the queue. A packet of P bits at the head of line in the buffer can move to the queue if the token counter has at least P tokens. In that case, P tokens are removed from the counter and the packet jump at once into the queue.

3.4.2 The Re-Queuing mechanism (RQM)

In the same scenario of Fig. 3.3, when the TB algorithm runs out of tokens, the out-of-profile EF and AF packets are detected, which means that the sender generates more packets per time unit than the packets allowed by the SLA. For fixed networks with a relatively low Round Trip Time (RTT), the general recommendation is to drop these out-of-profile packets [HBWW99]. Nevertheless, the situation over the DVB-S2 satellite link is different, given that in this type of link, the two-way propagation delay is significantly high.

In this condition, the proposal is based on considering not dropping the out-of-profile EF and AF packets, but instead send them to be en-queued again but in this case through the BE queue. This modified queuing mechanism is called the Re-Queuing Mechanism (RQM), which is also shown on Fig. 3.3. It is worth mentioning that in [RMMDA+12c] and [RMMDA+12d], an extensively performance evaluation have been conducted to test the improvements introduced by using the RQM mechanism for the AF and EF traffic classes respectively.

In this way, assuming that the protocol used to transport the EF and AF traffic classes is the TCP protocol. It is worth noticing that if the network layer drops a certain packet, the TCP receiver has to wait until it receives the retransmitted packet, before delivering the packets with higher sequence number to the application. Therefore, having a high RTT over the DVB-S2 link, increases the number of packets in the receiver buffer and those packets will have to wait a long period of time (around a RTT) to be delivered onto the application.

The main objective with the adoption of the RQM mechanism, it is to reduce the latency and jitter experienced by the user-application for the EF and the AF traffic classes. Given that our model does not drop these out-of-profile packets, but instead these packets are sent to the BE queue.

This modified RQM mechanism allows the out-of-profile packets to downgrade its QoS level, giving them the possibility (in the case of TCP) to reach the receiver before a RTT. The downgraded packet will get in the destination later than the other transmitted packets, since this downgraded packet will be re-classified to the BE traffic class. This packet disorder will be detected at the sender side by means of the TCP's triple dupACK mechanism. Therefore, the TCP will trigger the fast retransmit/fast recovery algorithm as a congestion control signal [MSMO97]. As a response the sender will reduce its congestion window, forcing to fulfill the SLA. It is worth mentioning that in this model the input traffic of the BE traffic class is not totally independent from its higher priority traffic classes.

Conversely, if the protocol used to transport high priority traffic classes is different to the TCP protocol, the ability to detect out-of-order delivery will depend on the upper-laying protocol (RTP) used to reconstruct the sender's packet sequences at the destination-user application.

3.4.3 The QoSatArt queue design

In the QoSatArt architecture, the TBs limiting rates are used to regulate the queue occupancy level to enforce the system to work at a reference load level. This reference working point is a function of the defined SLAs for each DiffServ traffic class. Hence, the design of the QoSatArt architecture requires setting up the queue size (B_i) for each specific traffic class (see Fig. 3.3) in order to keep the delay bounded for high priority traffic classes.

To do so, the *system load*, representing the number of packets in flight (when the system is transporting high volume of data traffic at time t), for each Diffserv traffic class i is expressed as:

$$L_i(t) = \mu_i(t) \cdot RTT_{min} + \beta_i(t) = BDP_i(t) + \beta_i(t) \tag{3.1}$$

Where $L_i(t)$ represents the *system load*, $\mu_i(t)$ represents the TB's limiting rates for each high priority traffic class i (in this case μ_{EF} and μ_{AF}), RTT_{min} represents the two-way propagation delay or minimum Round Trip Time (set to 560 ms in the GEO satellite scenario), $\beta_i(t)$ is the queue occupancy level and B_i is the queue size ($\beta_i(t) \leq B_i$, for all t).

Here, the maximum system load available for each traffic class i at time t is:

$$L_i^{max}(t) = \mu_i(t) \cdot RTT_{min} + B_i = BDP_i(t) + B_i \tag{3.2}$$

In order to allow the system to have enhanced link utilization, the queue length for the EF and AF traffic classes are set to their BDP values:

$$B_i = BDP_i(t) \tag{3.3}$$

However, in this scenario it is important to consider the average packet delay experienced at each queue called $Delay_i(t)$. Particularly, for the EF traffic class the experience average packet delay $Delay_{EF}(t)$ is a function of the error term (E_p) experienced for the treatments of individual EF packets [DCB+02]. Similarly, for the AF traffic class, this delay $Delay_{AF}(t)$ represents the experienced delay at each AF queue [HBWW99]. In both cases, the *gateway* is able to estimate E_p and $Delay_{AF_i}$ values for each traffic rate [10205].

Finally, for the BE traffic class, the queue length B_{BE} is set considering μ_{sat} representing the available bandwidth present in the bottleneck satellite link. This is carried out to allow the BE traffic class to use the remaining link bandwidth, according to the predefined *QoS policy*.

Therefore μ_{BE} and B_{BE} are set as:

$$\mu_{BE} = \mu_{sat} \tag{3.4}$$

$$B_{BE} = \mu_{sat} \cdot RTT_{min} = BDP_{sat} \tag{3.5}$$

By considering this queue design, it is possible to keep an optimal operation-working point while enhancing the satellite efficiency, given that the amount of packets to be buffered in the AQM system is set equal to the total in-flight packets that the satellite system is able to transport. Finally, when all packets have been queued and processed by the AQM module (see Fig. 3.3), they are sent directly to the IP scheduler.

3.4.4 The cross-layer design and the IP scheduler

The design of the QoSatArt architecture is developed over the SI layers of the ETSI-BSM-QoS protocol stack. This is done, in order to improve the management and control functions performed at the upper layers while enforcing the establishment of priorities among applications at higher layers [M M08].

The detail layered design, including its separation between the high SI layers and the low SD layers is shown on Fig. 3.4.

As it is shown, several elements are set up at the SI layers, which are mainly related to QoS provisioning based on the DiffServ framework: the *XL-Manager* module, the *QoS Manager*, the IP scheduler and the proposed cross-layer design between the physical and the network layers.

Here, the *XL-Manager* module is defined to manage and orchestrate the cross-layer interactions between SI and SD layers. In particular, this module is defined for calculating the cross-layer parameters to prioritize certain traffic classes according to the values reported by lower layers (see Fig. 3.4). In QoSatArt, the *XL-Manager* module works in a joint configuration with the *QoS Manager* and the IP scheduler.

The *QoS Manager* is defined to manage the QoS parameters for establishing priority levels and traffic rates according to the predefined SLAs. It considers the *QoS policy* defined in the QoSatArt architecture, in which the EF traffic class has the highest priority while the AF traffic class has more priority than the BE traffic class.

On the other hand, the *IP scheduler* is defined to provide traffic differentiation based on the QoS requirements. This element has an important impact on the support of QoS specifications [ETS03], mainly, because the IP scheduler defines how the *gateway* allocates the bandwidth capacity to those flows requiring higher QoS guarantees, at the forward channel [SA10]. Moreover, the *IP scheduler* controls the Diffserv queues to provide them with the suitable *per DiffServ class bandwidth* allocation [ETS03]. Here, it is important to mention that, given that the IP scheduler performs *per service* allocation which is not

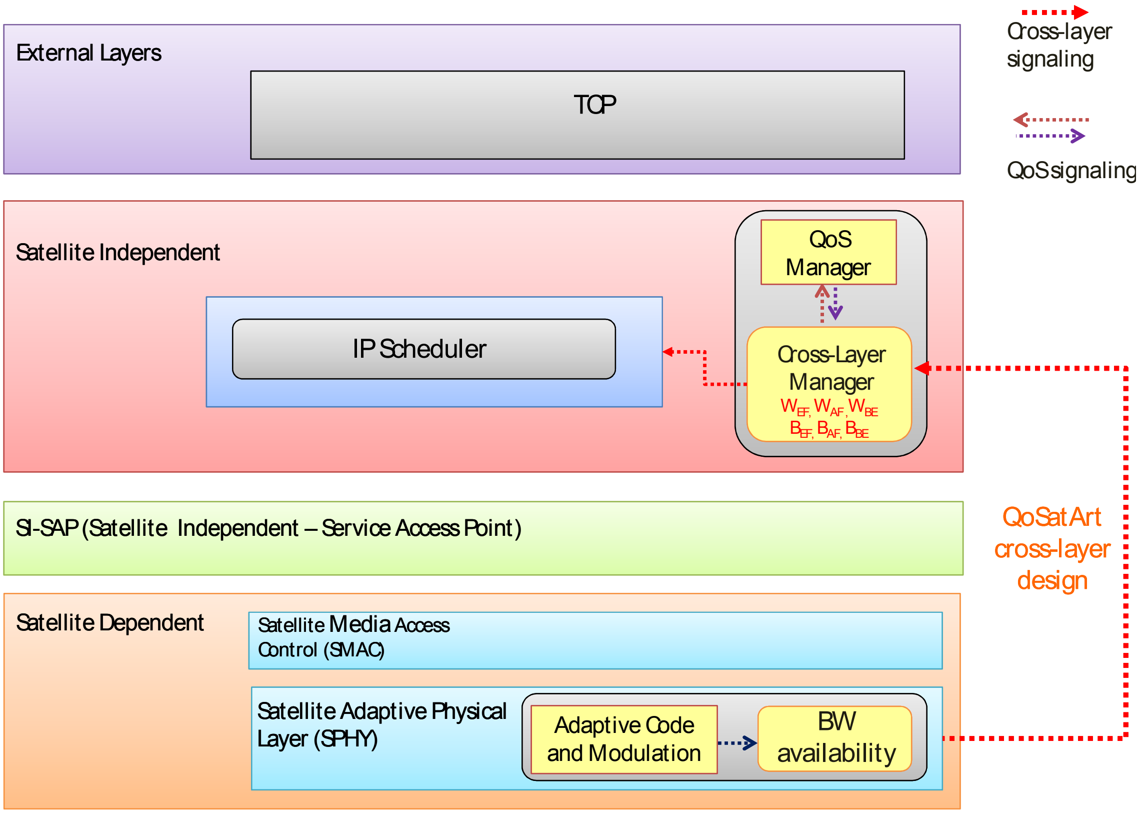


Figure 3.4: The cross-layer design for QoSatArt

uniformly distributed, the load sharing among physical queues is also not uniform. All packets coming from each traffic class are sent directly to the IP scheduler which controls the order in which packets are extracted out from its queues. Here, it is worth mentioning that the IP scheduler incoming rate is limited by each token bucket.

In particular, the design of the *IP scheduler* considers two factors to determine the minimum amount of bandwidth for each traffic class: the QoS specifications for each traffic class (specified by the satellite operator) and the available bandwidth present in a DVB-S2 satellite.

The QoS specifications are configured using the TB rate limiters. Here, the high priority TB-limiting rates are set to μ_{EF} and μ_{AF} for the EF and AF traffic classes respectively. However, the BE TB limiting rate is set to the maximum bandwidth availability C_{OUT} rate. This is done to allow the BE traffic class to use the remaining link bandwidth according to the predefined *QoS policy*. As a consequence, the IP scheduler incoming rate is controlled and limited by the TBs.

Using this design, it is possible to keep control of the incoming traffic load according to the SLA specifications. As a result, the main concern is to provide by means of the *IP scheduler* the correct prioritization of traffic considering the values reported by lower layers. In particular, the design at the SI layers takes into account, as input parameter the bandwidth availability (C_{OUT}) present in the satellite system. This value is determined considering the SNIR value reported by each RCST.

Here, a *cross-layer design* is proposed that considers the current capacity (C_{OUT}) of the DVB-S2 satellite link (considering the ACM adaptation behaviour defined in the DVB-S2) as the main criteria for the definition of QoS priorities to provide enhanced allocation of the satellite network resources. This cross-layer mechanism is proposed between the physical layer and the network layer to enhance the provisioning of the QoS guarantees established in the SLA. In the view of the author, this is the first time a cross-layer mechanism based on the SI layer (*SI-centric approach*) that used the ACM adaptation behaviour as a main criteria to prioritize traffic classes over BSM satellite systems, is proposed.

Fig. 3.4 shows the proposed *cross-layer design*. Here, it is worth observing that the *IP scheduler* is decoupled from the *XL-Manager* module. This is done in order to reduce the scheduler complexity while allowing that both modules can work independently based on their own settings.

One of the main contributions of the QoSatArt architecture, it is that the *IP scheduler* is able to guarantee the QoS requirements for specific traffic flows using a single parameter: *the bandwidth availability*. This is done, by applying the proposed cross-layer design that considers the physical layer changes due to the ACM adaptation behaviour. This design allows to provide QoS guarantees among DiffServ flows taking into account the link bandwidth variations reported by the physical layer.

3.4.5 QoSatArt signaling mechanism

The proposed cross-layer optimization requires adding QoS signaling mechanisms to enable the interchange of information between layers and functional blocks. One of the main functions related to QoS signaling is the definition of mechanisms used to carry information about the requested Service Level Specification (SLS) through layers.

The Service Level Agreement (SLA) defines the set of QoS parameters available to a customer that determines the minimum QoS requirements of the connection. Therefore, an SLA variation are handled by a SLA negotiation. It implies modifications inside mechanism that provide QoS guarantees. These changes are performed by the QoS signaling mechanism.

The design of QoSatArt architecture assumes that the satellite operator is able to provide signaling and managing functions based on the SLA modifications. These functions can be performed by either human operators or applications that update the QoS requirements defined by users. This requires previous authorization from the Network Control Center (NCC). Particularly, in the case of interactive services using the return channel, the RCST terminal requests an application-specific service by sending a *Service request* to the NCC. The *Service Controller* inside the NCC is in charge of determining the QoS needs of the requested service.

In QoSatArt architecture, the QoS signaling messages are sent using out-of-band signaling. Here, the the Single Network Management Protocol (SNMP) is proposed to transport

QoS signaling. The SNMP protocol consists of a set of standards for network management, including an application layer protocol, a database schema, and a set of data objects. It has been standardized by the IETF in the RFC 3411 [HPW02], which has been commonly used on networking devices with open source implementations.

An SNMP-managed network consists of three main elements: the network management system (NMS), the managed device and the agents.

A network management system (NMS) is an external element that executes applications to monitor and control agents allocated inside managed devices. NMSs provide the bulk of the processing and memory resources required for network management. Particularly, in QoSatArt architecture the NMS module is allocated inside inside the *Bandwidth Broker* as a part of the NCC.

The agents are allocated inside the Hub (for the forward channel) and the RCST (for the return channel). Both of them implement SNMP interfaces that allow the access to node-specific information. The agents are network-management software modules that have local knowledge of management information and translates that information to or from an SNMP specific form.

Given that SNMP itself does not define which variable should be monitored, it is possible to use an extensible design, where the available information is defined by the management information bases (MIBs) [Pre02]. MIBs describe the structure of the management data of a device subsystem.

In QoSatArt architecture, the cross-layer information, representing the changes in the bandwidth availability C_{OUT} generated by heavy rain event, are stored in the MIBs.

Fig. 6.3 illustrates the QoSatArt signaling mechanism. As it is shown the *SNMP manager* monitors and manages the network devices at the Control plane (C-plane). It interacts with the *SNMP agents* to report or request information. In this way, the *SNMP manager* requests information from the *agent* using `Get-Request` message, allowing the *agent* to respond using the `Get-Response` message.

Alternatively, if an event occurs on the managed device, the *agent* is able to send a `Trap` message to the manager, indicating the variables that have changed.

In particular, the *SNMP manager* periodically polls the bandwidth availability C_{OUT} stored in the MIB (provided by the Physical layer). This value is get by the *XL-Manager* using a `Get-Request` message to request the Physical layer when a modification of the available bandwidth occurs in the satellite system. Here, a *Response* message indicating the updated values from the *SD layer* is returned.

3.4.6 The Satellite Dependent design

The SD layer design inside the *gateway* is shown on Fig. 3.2. As it is observed, packets are sent through the SD layer (after scheduling functions) using the QoS mapping concept [10207]. This implements Queue Identifiers (QID) to send the DiffServ flows from the SI layers to and the SD layers. This procedure is allocated at the Satellite Independent - Service Access Point (SI-SAP) interface, which follows the guidelines defined in [M M08].

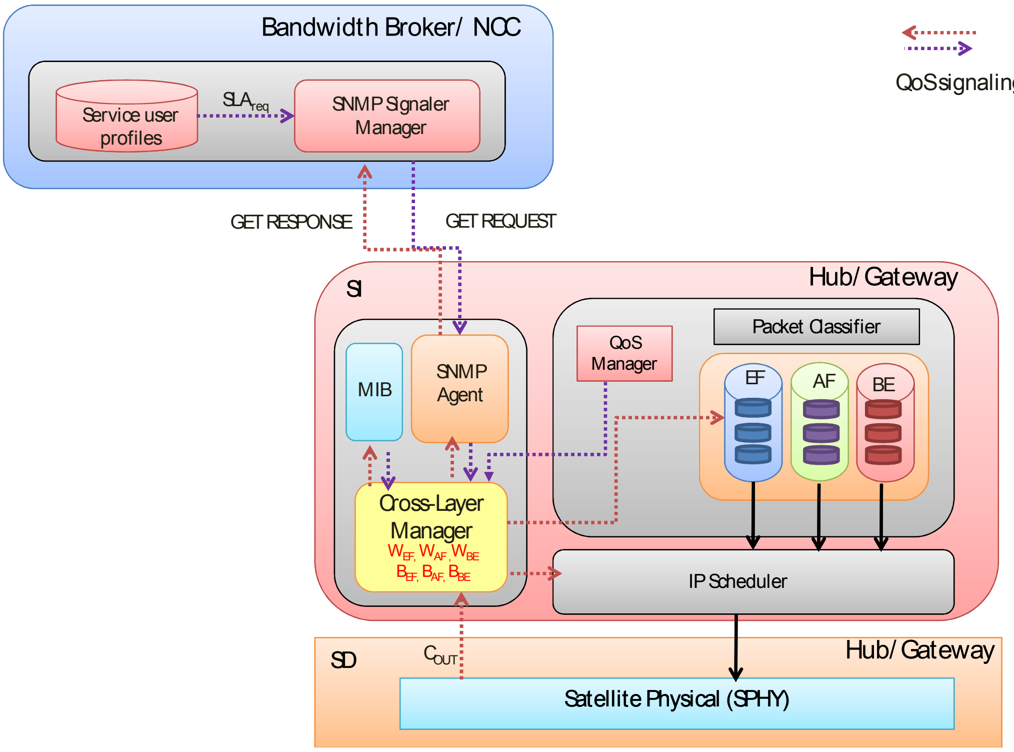


Figure 3.5: XPLIT Signaling Mechanism.

At this point, each QID enables the IP packet to be allocated to a virtual queue (considering its QoS characteristics) and then transported across the SD layers. For simplicity the QoS mapping between SI-SD layers is not detailed on Fig. 3.2.

The SD layer design includes a set of physical buffers, their associated Encapsulation Units, a Frame Scheduler and a DVB-S2 multiplexer. These elements are used to construct frames using IP packets and transmit them through the DVB-S2 physical layer to those destination terminals that have similar propagation conditions.

In particular, when enough packets are stored in a *ModCod* queue, the Encapsulation Units are used to build DVB-S2 frames which are served by the Frame Scheduler to feed the satellite physical layer (see Fig. 3.2).

According to the DVB-S2 frame structure, frames coming from different encapsulating units are transmitted through the satellite link using different *ModCods*. The corresponding *ModCod* is determined by the terminal that is under the worst propagation conditions, assuming that all destination terminals (in a beam) have similar propagation conditions. As a consequence, the frames sent to terminals having good propagation conditions (i.e. in clear sky) will be queued to encapsulation units using a $ModCod_i$ that provides low bit protection. Conversely, the frames sent to terminals having bad propagation conditions (i.e. facing a rain event) will use a $ModCod_n$ with higher bit protection levels, thus requiring additional overhead.

Here, the DVB-S2 frame length (payload) is assumed to have a constant length (between 16200 and 64800 bits) and each frame can encapsulate several IP packets. The physical queues have enough size to store at least two of these frames.

Using Generic Stream Encapsulation (GSE), frames are encapsulated and sent to the frame scheduler which is a lower level scheduler responsible for dealing with *fairness*. Here, a basic frame scheduler strategy is recommended [ECB08] to achieve acceptable overall performance in terms of low response time and high spectral efficiency. As the frame scheduler should guarantee a maximum waiting time for each IP packet while implementing the suitable *fairness* policy among terminals.

To guarantee a maximum waiting time for each IP packet, the QoSatArt frame scheduler follows a strategy based on the queue status [ECB08]. In this way, if an IP packet has reached its maximum waiting time and it is still waiting in the queue (of a certain encapsulation unit), the frame scheduler should encapsulate this IP packet into the next DVB-S2 frame (considering a *ModCod* with higher protection level) and transmit it through the satellite link. However, if the minimum frame size is not reached, several IP packets from other encapsulation unit (having different $ModCod_j$ that require less protection) must be used to fill the frame size. As a result it is possible to assure the maximum waiting time for each IP packet.

Finally, to guarantee *fairness* among terminals, the QoSatArt design (as most of the broadcast systems) is based on the Time Division Multiplexed (TDM) sharing policy with the adoption of ACM techniques [PAPC07]. In this context, terminals under good propagation conditions are able to use *ModCods* with lower overhead, allowing to transmit them with higher rates compared to those terminals under bad weather conditions. For this reason in practice, the *fairness* policy applied to the DVB-S2 frame scheduler tries to shield the network layer from the effects of a time and location-dependent physical layer [FVG06].

In this way, using ACM techniques it is possible to select the proper *ModCod* to guarantee a determined (low) error probability [30097], [MM06]. Therefore, to offer homogeneous service among terminals with similar propagation conditions, the DVB-S2 frame scheduler approximately assigns the same shared-service rate. As a result, the offered transmission rate is the same, while the time used to transmit frames depends on the propagation conditions for each destination terminal. Notice that if the frame scheduler implements other policy like sharing the same amount of transmission time among terminals, it will penalize the terminals under the worst propagation conditions, so a similar situation would be present to what happen within DVB-S.

3.5 Conclusions

In this chapter the QoSatArt architecture developed to provide QoS guarantees for IP traffic over the DVB-S2 satellite channel has been presented. The QoSatArt is designed in compliance with the ETSI-BSM-QoS framework. Here, a detailed design at the Satellite

Independent (SI) layers in order to provide QoS guarantees by means of traffic differentiation has been proposed.

A cross-layer optimization between the physical layer and the network layer is proposed to guarantee the QoS requirements established in the SLA. Here, the main consideration is that the satellite system is affected by rain events which reduce the bandwidth availability.

The architecture includes the design of the Active Queue Management (AQM) system to minimize the delay values experienced at the Application layer for delay sensitive traffic. Here, a complete QoS design inside the *gateway* has been detailed, in which the RQM mechanism is proposed to enhance the goodput for the EF and AF traffic classes while reducing the end-to-end delay and jitter.

In addition, a cross-layer IP scheduler is proposed to guarantee the high priority traffic classes regardless of the channel condition affected by rain events. The proposed architecture provides low complexity on its implementation and seamlessly inter-operates with terrestrial IP networks.

The next chapter is developed to specify the IP scheduler design together with the algorithm developed to calculate the weighted values to provide QoS guarantees. In addition, the simulation results of evaluating the proposed QoSatArt architecture are presented.

CHAPTER 4

QoSatArt application case: QoS support based on IP scheduling

The support of QoS guarantees based on a packet scheduler has been primarily investigated in the context of wired networks with constant link capacity. Here, different criteria has been proposed to calculate scheduling values aiming to differentiate certain traffic classes. Parameters such as the mean packet length, queue sizes, average packet delay and packet losses have been taken into account.

For instance, in [ITI02] a Variable Weighted Round Robin (VWRR) scheduler is proposed, this design uses the average packet length to adapt its weighted values. Similarly, in [WWSS01] the problem of adjusting the weights of the WRR scheduler to guarantee the premium service has been analyzed. Here, the authors propose a resource allocation model that considers the average queue size to prioritize traffic. In particular, the queue size is calculated using a low-pass filter.

Moreover, in [YDKD02] a modified Fair WRR (FWRR) scheduler has been proposed to protect the best effort traffic from the assured forwarding out-of-profile packets in the core routers. This proposal is based on the DiffServ architecture that dynamically adjusts the weights and the buffer allocation using congestion hints to avoid unfair bandwidth sharing.

In [DSR02] the authors analyze a case to provide specific type of relative services based on a Proportional Differentiated Service (PDS) model. Such work includes a description of packet scheduling algorithms to get an enhanced approximation to the proportional differentiation model. The authors conclude that the Waiting Time Priority Scheduler (WTP), which increases the priority of a packet proportional to its waiting time, provides a suitable approximation to PDS.

Taking a different approach, other works reported in the literature are focused on QoS differentiation based on a revenue criteria [SHJK06], in which classes with higher prices are prioritized to get a better QoS level. In [SHJR04], an adaptive scheduling scheme using the revenue-based WRR is presented. This proposal has been considered for guaranteeing the maximum service provider revenue. The proposed scheme, adjusts its weights using the revenue criteria to control the resource allocation. Similarly, authors in [AZ12] propose a game theory approach to be applied over wireless networks in order to provide enhanced resource allocation. Here, a comprehensive cross-layer framework including physical and medium access control layer requirements are proposed to provide fair distribution of resources with a minimum fairness index.

Taking a different perspective, there are other works focussed on the design of a packet scheduler to provide service differentiation based on a cross-layer mechanism. Particularly, the scheduler design based on the Proportional Differentiated Services (PDS) model has been addressed by the authors in [CKP09]. Here, the provisioning of a proportional class-based service differentiation to TCP flows in the context of split-TCP scenarios is proposed. The scheduling algorithm for controlling the resource allocation is set up at the MAC layer in the context of a Bandwidth on Demand (BoD) satellite scenario. This proposed scheduler is based on the WTP scheduler which considers the average queuing delay to provide service differentiation. In addition, it proposes the use of Performance Enhanced Proxies (PEPs) to configure variables for different TCP connections. This scheduler is allocated at MAC layer and the PDS model is also implemented at TCP layer based on the throughput for each TCP flow.

Similarly, in [CG07] a cross-layer design for a packet scheduler over a Multi-beam Broadband Satellite (MBS) system is proposed. The scheduler is set up at the MAC layer which considers the physical layer behaviour of the Ka band satellite propagation channel. Here, the scheduler determines the fraction of time assigned to each physical layer for transmission by considering a correlated area where the users undergo similar channel conditions. Similarly, in [LYZ$^+$10] a distributed MAC scheduler for wireless mesh networks is proposed to offer QoS guarantees in a heterogeneous environment. Here, a QoS-aware fair packet scheduling (QFPS) algorithm to fulfill the QoS provisioning is presented.

In this chapter, the QoSatArt rules defined in Section 3.4.4 are applied in order to specify the detailed design of the IP packet scheduler for supporting QoS guarantees. Here, an algorithm inside the *XL-Manager* is proposed to calculate the associated values (weighted) aiming to enhance the differentiation of high priority traffic classes. The proposed algorithm performs an adaptation based on the intensity of rain events [CGR08], which is constantly updated using a cross-layer design between the physical layer and the network layer. This is done to provide QoS provisioning when different levels of link capacity are available in the satellite system.

4.1 The cross-layer IP scheduler design

The IP scheduler considers two factors to determine the minimum amount of bandwidth for each traffic class: the QoS specifications for each traffic class (specified by the satellite operator) and the available bandwidth present in a DVB-S2 satellite.

Particularly, the QoS specifications are defined using the TB rate limiters. Here, the high priority TB-limiting rates are set to μ_{EF} and μ_{AF} for the EF and AF traffic classes respectively. However, the BE TB limiting rate is set to the maximum bandwidth attainable C_{OUT} rate. This is done to allow the BE traffic class to use the remaining link bandwidth according to the predefined QoS policy. As a consequence, the IP scheduler incoming rate is controlled and limited by the TBs.

Using this design, it is possible to keep control of the incoming traffic load according to the SLA specifications. As a result, the main concern is to provide by means of the packet scheduler the correct prioritization of traffic considering the bandwidth availability present in the DVB-S2. For this purpose, we assume that the corresponding weighted values established to prioritize the resources for each traffic class are integer values defined by W_{EF}, W_{AF} and W_{BE} for the EF, AF and BE traffic classes respectively. The cross-layer design proposed in this paper actively adjusts these values to enforce its QoS level in the presence of a heavy rain event.

In this way, considering the three traffic classes defined in the DiffServ architecture (EF, AF and BE), the proposed algorithm will therefore prioritize the EF traffic class over the AF and BE traffic classes when the satellite system experiences a capacity reduction in the presence of a heavy rain event. This will be carried out by assigning high weighted values to the EF traffic class while the AF and BE traffic classes will have the lowest weighted values [WWSS01].

Similarly, when the intensity of the rain event has diminished and the link capacity has increased, the algorithm will keep the priority levels of the EF and AF traffic classes over the BE traffic by assigning high weighted values to the EF and AF traffic classes while the BE traffic class will have the lowest value.

Finally, when the rain event has ended and the bandwidth availability is enough to guarantee not only priority traffic (EF and AF), but also there is remaining bandwidth for the BE traffic class, the weighted values will be assigned according to the rate of the high priority classes. Here, the adoption of the Proportional Delay Differentiated (PDD) model has been proved to successfully provide service differentiation to networks with heavy load conditions [CKP09].

In order to specify the algorithm to provide QoS priorities, the adaptive distribution of the available capacity among each traffic class is proposed. This is done, using three capacity intervals (*Scarce, Constrained and Broad*). These intervals are defined considering the parameters specified in the SLA, which the satellite operator is able to modify according to its requirements. Fig. 4.1 shows the weighted values calculated at each capacity interval as a function of the DVB-S2 system capacity (C_{OUT}).

4.1.1 The scheduler algorithm

As it is observed on Fig. 4.1, the three capacity intervals are established to enhance the QoS differentiation when a reduction of bandwidth capacity is experienced. Here, the proposed intervals are delimited by the high priority TB-limiting rates as follows (see Fig. 4.1): the $UpperCap_{THR}$ threshold represents the value when the satellite capacity is reduced up to the sum of the EF and AF TB-limiting rates (see Eq. 4.1) Similarly, the

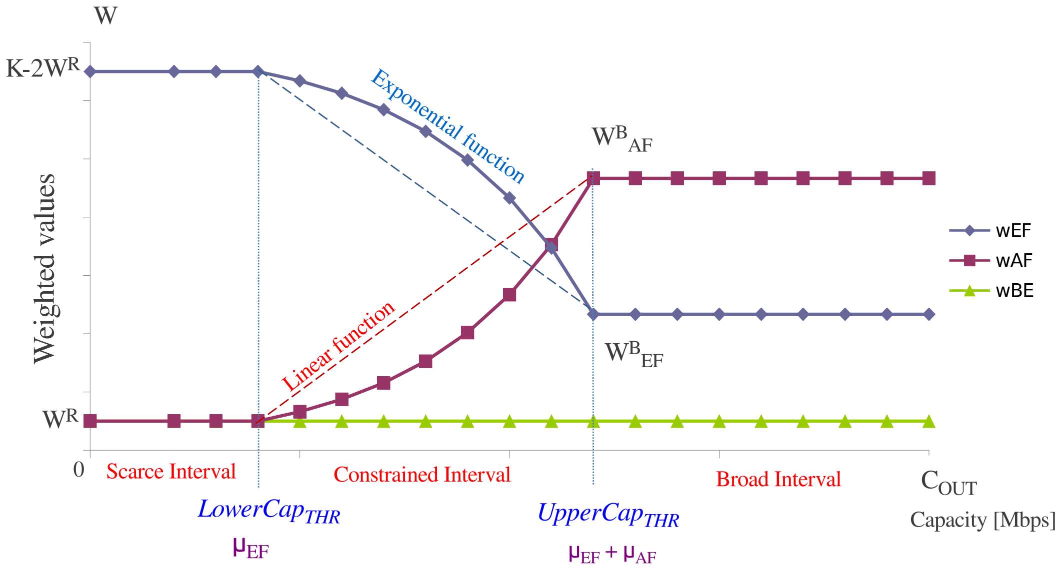


Figure 4.1: The weighted values as a function of the DVB-S2 system capacity (C_{OUT})

$LowerCap_{THR}$ threshold represents the value when the satellite capacity is affected up to the EF TB-limiting rate (see Eq. 4.2).

$$UpperCap_{THR} = \mu_{EF} + \mu_{AF} \tag{4.1}$$

$$LowerCap_{THR} = \mu_{EF} \tag{4.2}$$

Following equations 4.1 and 4.2, the *Scarce Capacity interval* specifies the situation in which the system capacity (C_{OUT}) is scarce but it is still enough to guarantee part of the EF traffic class. Therefore, it is delimited by:

$$C_{OUT} \leq LowerCap_{THR} \tag{4.3}$$

Similarly, the *Constrained Capacity Interval* defines a situation in which the system capacity is enough to totally guarantee the EF rate but partly guarantee the AF traffic class. It is limited by:

$$LowerCap_{THR} < C_{OUT} < UpperCap_{THR} \tag{4.4}$$

Finally, the *Broad Capacity interval* considers the situation in which the available bandwidth in the system is broad enough to totally transport both high priority traffic classes, while assigning the remaining link bandwidth to the BE traffic class, this interval is considered if:

$$C_{OUT} > UpperCap_{THR} \tag{4.5}$$

According to the *QoS policy*, the BE traffic class should to use the remaining link bandwidth. Therefore, the proposed algorithm allocates the excess of resources on the BE class when the system capacity is working in the *Broad Capacity interval*. This is done by considering the BE TB limiting rate which is set to the maximum bandwidth availability C_{OUT}.

However, given that the BE class has the lowest priority level, the algorithm reduces the BE bandwidth assignment when the system capacity is working in the other intervals. Therefore, for all the cases, W_{BE} is set to W^R, representing a minimum and constant value called residual weight, see 4.6.

$$W_{BE} = W^R \tag{4.6}$$

Using this value (W_{BE}) in combination with the token bucket limiting rate, it is possible to allocate the remaining link bandwidth to the BE traffic class, while reducing its participation when the system capacity is working in the other intervals, meanwhile, the bandwidth for high priority traffic classes is guaranteed.

On the other hand, to allow a trade-off between the bandwidth assignment among each traffic classes while providing service differentiation, the associated weighted values are set to K (Eq. 4.7). Where, the selected K is considered a constant and integer value that should verify an accuracy level. Here, the sum of the weighted values (W_{EF}, W_{AF} and W_{BE}) is set as follows:

$$W_{EF} + W_{AF} + W_{BE} = K \tag{4.7}$$

Particularly, for the cases when the system capacity is in the *Broad Capacity interval* (see Fig. 4.1), the algorithm will assign constant weighted values to guarantee the priority classes (EF and AF). At this interval, the weighted values for each traffic class are set to W^B_{EF} and W^B_{AF} for the priority classes and W^R for the BE traffic class. Following this assumption the next relationship is obtained:

$$W^B_{EF} + W^B_{AF} = K - W^R \tag{4.8}$$

Hence, in order to define the relationship between de associated weighted values and the TB-limiting rates (for the high priority traffic classes). The algorithm considering the proportional distribution of the system resources is applied, following the specifications defined in [CKP09]. Then, the ratio of W^B_{EF} and W^B_{AF} is set equal to the ratio of the TB-limiting rates.

$$\frac{W_{EF}^{B}}{W_{AF}^{B}} = \frac{\mu_{EF}}{\mu_{AF}} \tag{4.9}$$

Here, it is worth mentioning, that selected K value represents a constant and integer value. However, the division $\frac{\mu_{EF}}{\mu_{AF}}$ has not necessarily an integer outcome. Hence, according to the proportional distribution, the K value should verify Eq. (4.10), where a represents the accuracy level it is desired for the selected weighted values:

$$\frac{\mu_{EF}}{\mu_{AF}} - \frac{W_{EF}^{B}}{K - W_{EF}^{B} - W^{R}} < a\% \tag{4.10}$$

Given that W_{EF}^{B} and W_{AF}^{B} are integer values, these can be expressed as a function of the TB-limiting rates (μ_{EF} and μ_{AF}) as:

$$W_{AF}^{B} = \left\lfloor \frac{K - W^{R}}{\mu_{EF} + \mu_{AF}} \mu_{AF} \right\rfloor \tag{4.11}$$

$$W_{EF}^{B} = \left\lceil \frac{K - W^{R}}{\mu_{EF} + \mu_{AF}} \mu_{EF} \right\rceil \tag{4.12}$$

Until now, the application case has been solved for the *Broad Capacity interval* (see Fig. 4.1). Nevertheless, when the system capacity is in the *Scarce Capacity interval*, the proportional distribution (PDS) algorithm is not able to guarantee the high priority traffic classes. This is mainly because the proportional values are calculated considering clear sky condition. Hence, a second algorithm is proposed, which assigns constant weighted values to guarantee the EF priority class over the AF and BE traffic classes in order to provide the highest priority o the EF traffic class W_{EF}:

$$W_{EF} = K - 2W^{R} \tag{4.13}$$

As a result the W_{AF} and W_{BE} should be provided with the minimum weighted value which are set as:

$$W_{AF} = W_{BE} = W^{R} \tag{4.14}$$

Finally, the application case defined to guarantee the EF traffic class over the AF traffic class is presented. It covers the cases when the system capacity is in the *Constrained Capacity interval*. Here, the proposed algorithm should allocate most of the resources to the EF traffic class, providing them with the highest weighted values. Therefore, to link the thresholds $UpperCap_{THR}$ and $LowerCap_{THR}$ (see Fig. 4.1), in order to have a continuous curve, it is proposed to apply either a linear or an exponential relationship (see Fig. 4.1) between the weighted values of the higher priority classes (W_{EF} and W_{AF}) and the satellite link capacity (C_{OUT}). Therefore, to define an exponential function at the *Constrained Capacity interval*, the W_{AF} and W_{EF} values are defined using the following equations:

$$W_{AF} = W^R \left[\frac{W_{AF}^B}{W^R} \right]^{\frac{C_{OUT} - \mu_{EF}}{\mu_{AF}}} \tag{4.15}$$

$$W_{EF} = K - W^R - W_{AF} \tag{4.16}$$

Similarly, to define a linear function at the *Constrained Capacity interval*, W_{AF} and W_{EF} are defined using the following equations:

$$W_{AF} = \frac{W_{AF}^B - W^R}{\mu_{AF}} \times C_{OUT} + W^R \tag{4.17}$$

$$W_{EF} = \frac{W_{EF}^B - K + 2W^R}{\mu_{AF}} \times C_{OUT} + K - 2W^R \tag{4.18}$$

It is worth mentioning that in [RMMDA+12b] and [RMMDA+12a], a practical analysis performed between the high priority traffic classes and the available bandwidth at the *Constrained capacity interval.* Such analysis has been presented in the simulation results Section 4.3, in which the cross-layer (weighted) values at the *constrained interval* have been calculated considering two functions: a *linear function (XL-linear)* and an *exponential function (XL-exponential).*

4.2 Simulation environment

4.2.1 Satellite network settings

The simulation is carried out employing the NS-2 simulator tool version 2.29. For QoS purposes the DiffServ module was developed using the library described in [PEBS00] in which the proposed cross-layer design and the algorithm for the IP scheduler has been added.

The simulated network topology used to test the cross-layer packet scheduler is shown on Fig 3.1. The transport protocol defined for the AF and BE traffic classes is the Sack [BAFW03] variant of the Transmission Control Protocol (TCP). We used the TCP Linux version [DP06], which includes this TCP variant among others. The EF traffic class is composed using User Data Protocol (UDP) datagrams to strictly guarantee bandwidth reservation.

Here, it is worth mentioning that the main objective of the proposed simulation scenario is to evaluate the cross-layer IP scheduler, when extreme propagation conditions are experienced in the satellite system. In particular, when an extreme reduction of bandwidth capacity caused by a heavy rain event is present. Therefore, the satellite channel capacity, representing the bottleneck link, is set considering that terminals allocated in clear sky conditions have 3.4 Mbps of bandwidth. Alternatively, terminals allocated under a heavy rain event experience a bandwidth reduction down to 0.6 Mbps. Therefore, the satellite link capacity C_{OUT}, represented by the sinusoidal wave, fluctuates between 0.6

and 3.4 Mbps as it is shown on Fig. 4.2. The model that allows the characterization of the physical layer behaviour follows the specification defined in Section 4.2.2.

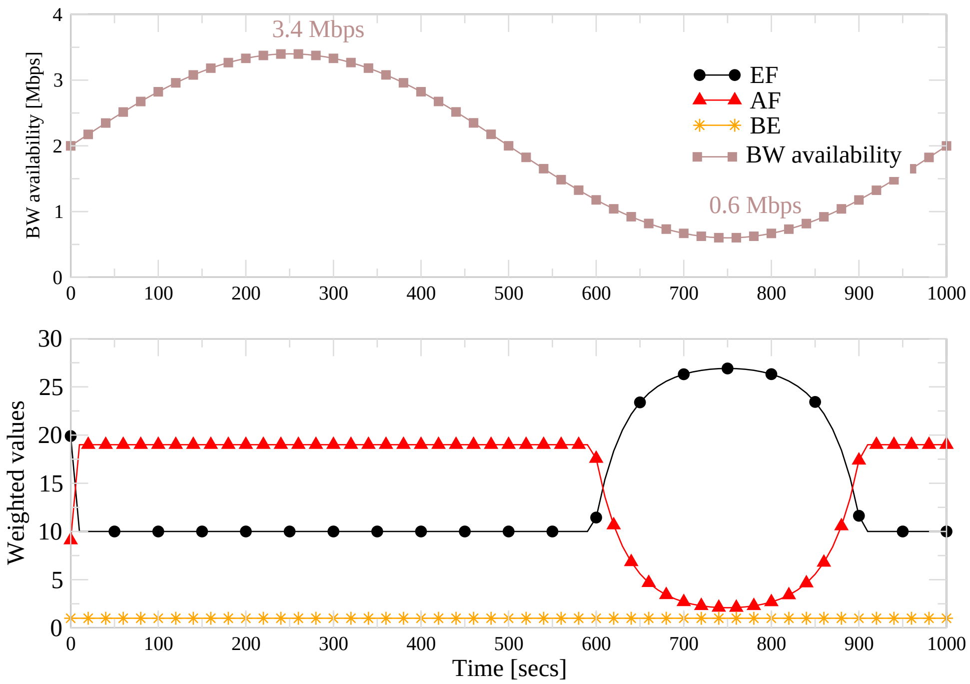


Figure 4.2: Simulated weighted values (lower graph) using the exponential algorithm as a function of the bandwidth availability (upper graph).

The DVB-S2 satellite network settings used throughout the simulation are configured as follows: The minimum Round Trip Time (RTT_{MIN}), considered as the two-way propagation delay value, is set to 560 ms. The buffer length of each traffic class is set to the BDP value that yields 90 packets. The forward link is based on the DVB-S2 standard which uses the ACM scheme to achieve lower Packet Error Rate values. To perform the simulation we consider a Bit-Error-Rate value set to BER $= 10^{-7}$. The Committed Information Rate (CIR) for the EF, AF and BE token buckets are set to 0.4 Mbps, 0.8 and 3.4 Mbps respectively. The Maximum Transmission Unit (MTU) size is set to 1500 bytes. The DVB-RCS link has permanent bandwidth capacity set to 256 Kbps. We also assume that there is no loss of ACK messages and that such messages have more priority level than data packets when they are sent through the RCS terminal.

The simulation parameters (number of flows, transmission rate and C_{OUT}) for each service class are presented on Table 4.1. Importantly, the BE traffic class will use the remaining bandwidth. The $LowerCap_{THR}$ and the $UpperCap_{THR}$ values are set to 0.6 Mbps and 1.4 Mbps respectively according to equations 4.1 and 4.2.

4.2.2 Satellite bandwidth characterization

In order to determine the values of the transmission rate variations at the physical layer, which are the input parameters of the proposed cross-layer design, we have studied the case in which the bandwidth availability in the satellite system is reduced by a heavy rain event.

The recent standards developed for the DVB-S2/RCS physical layer [ets04] [ets05a] define as normative the use of the ACM [RVCM04] techniques to attain Interactive Services. One of the main advantages of using the ACM techniques is the ability to achieve quasi-error free channel conditions for each individual user, by providing them with the most suitable Modulation and Code (*ModCod*) value according to the measured SNIR, so that the spectral efficiency is as high as possible in all the cases. To do so, the DVB-S2 physical layer takes advantage of the Signal-to-noise-plus-interference-ratio (SNIR) value reported by each RCST [RG04].

As it is well recognized, the presence of propagation losses and transmission rate variations in the satellite link are mainly caused by atmospheric conditions such as the rain fade effects, which are the most common phenomena in Ka-band (20-30 Ghz) satellite systems. These events can lead to a Ka channel attenuation ranging from a few dBs up to more than 20 dBs. In addition, the maximum rain attenuation experienced in Ka systems has a difference about 5 dB (of peak reduction) between the attenuation affecting the best and worst locations [RVCM04],.

According to [ECC02], trying to describe and simulate a rain fade distribution using a simple mathematical function becomes a difficult task. Mainly because, the Power Spectral Density (PSD) shape of rain fade is not only time variant (varying from a few seconds to some hours), but also its duration is not determined by the same phenomena. For instance, short durations are mainly due to scintillation and multiple scattering while long durations are caused by the space-time rain structure. Several approaches of Ka-band rain fade have experimentally observed that the PSD of this process can be approximated by a low-pass filter. In particular, authors in [CGR08] propose a model that estimates the time variant channel reported for a heavy rain scenario in the context of the DVB-S2 forward link. Here, a DVB-S2 physical layer adaptation considering ACM techniques is performed, taking advantage of Channel State Information (CSI) reported by each RCST terminal. Such design considers two low-pass filters, one for characterizing the rain attenuation and a second filter for the atmospheric scintillation. In addition, the design rules for setting the hysteresis thresholds are also presented.

In this way, we propose to assess the physical layer performance using this channel estimation model. Here, we have chosen several points of the estimated SNIR curve representing the ACM adaptation behaviour for a heavy rain event [CGR08]. Given this we perform a statistical regression in order to determine the relationship between each of the selected points and its corresponding estimated bandwidth. As a result, we obtained a bandwidth fluctuation wave that varies between 3.6 to 4.4 Mbps, following the distribution shown on Fig. 4.3 (black line). These calculations consider a TDM DVB-S2 forward channel with eight *ModCods* combinations (QPSK (1/3, 1/2, 2/3 and 3/4), 8PSK (2/3 and 3/4) and 16-QAM (2/3 and 3/4)) in which the resulting data rate fluctuates between 51 Mbps to 204 Mbps considering 46 satellite antenna beams.

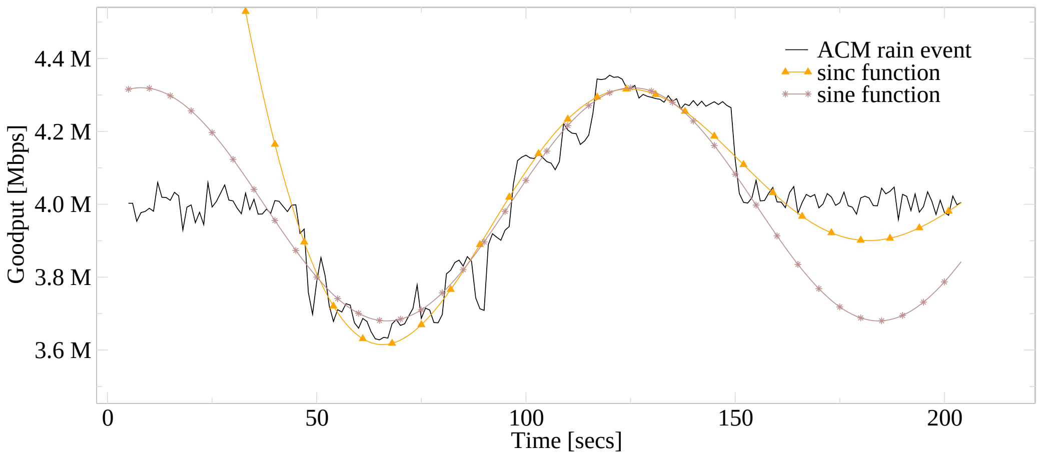


Figure 4.3: The ACM bandwidth characterization affected by a rain event

Keeping this idea, we have approximated the results by matching the bandwidth fluctuation (obtained by regression) using several mathematical functions. Therefore, the particular rain event (black line) can be experimentally fitted by using a sinc function, as it is shown on Fig. 4.3 (see symbol $\triangle$). Similarly, we have also fit the physical layer estimation by means of a sinusoidal wave (see Fig. 4.3 symbol $*$) to represent the same rain event affecting the DVB-S2 forward satellite link [RMMDA$^+$12d]. In this way, the peak of the sinusoidal wave, representing high link bandwidth availability will be set to 4.3 Mbps, while the valley representing a heavy rain event, will decrease the total link capacity to 3.6 Mbps. As a result, the bandwidth capacity will fluctuate between the minimum and maximum value of the sinusoidal wave.

Although the proposed functions do not perfectly match the ACM rain event as it suffers from intensive fluctuations, we believe that the proposed sinusoidal approximation allows to represent the general situation in which the satellite capacity is reduced. Therefore for simulation purposes, a single period (the valley) is enough to represent a deep reduction of capacity experienced in the satellite system.

4.2.3 Performance metrics

To carry out the performance evaluation of the selected TCP variants the following metrics are considered:

(i) The *goodput* represents the average amount of correct received data (excluding retransmissions) which is measured during a certain period of time. Particularly, as the satellite bottleneck link is set to 2 Mbps, the reachable goodput value for each traffic class must be in accordance with the SLA example previously defined. In this way, we expect to reach 400 Kbps of goodput for the EF traffic class (using UDP), 800 Kbps for the AF traffic class and 800 Kbps for the BE traffic class (using any TCP variant).

(ii) The *queue occupancy* metric represents the fullness level of each QoS queue which also determines the system latency. One of the main goals in any satellite system is to reduce the system latency. This is carried out by minimizing the buffer occupancy levels since the transmission path delay is very high and cannot be minimized. Therefore, we would rather employ a TCP variant that improves low levels of queue occupancy.

(iii) The *one-way delay* metric represents the time it takes a packet to go from the *gateway* to the remote RCST terminal. It includes the amount of time that a destination system spends processing the packet. Using QoSatArt architecture, it is possible to guarantee reduced per-packet delay values.

(iv) The *jitter* metric represents the difference in the E2E one-way delay between consecutive packets traveling in a flow with any lost packets being ignored. The effect is also referred to as delay variation.

4.3 IP scheduler algorithm: performance results

In order to evaluate the performance of the proposed cross-layer scheduler, three simulation tests are considered: the first test is performed to assess the DVB-S2 system response when the Round Robin (RR) mechanism is employed.

The second test considers the Weighted Round Robin (WRR) mechanism with static weight values set to 6, 8 and 1 (K = 15) for W_{EF}, W_{AF} and W_{BE} respectively. These values are set considering the proportional distribution of resources [CKP09]. Particularly, a higher weight value to the EF traffic class has been assigned than its corresponding proportional value. This is done because its proportional value is not able to guarantee the priority level for the EF traffic class when the capacity is reduced.

Finally, a third simulation test is performed considering the proposed cross-layer design (defined in Section 3.4.4) and the IP scheduler to enforce the priority levels taking into account the link capacity variations (defined in Section 4.2.2). Here, the cross-layer (weighted) values at the constrained interval have been calculated considering two functions: a linear function (XL-linear) and an exponential function (XL-exponential) (defined in Section 4.1.1).

The simulation results consider the XL-exponential and the XL-linear algorithms. The bandwidth rate for each traffic class is set as it is shown in Table 4.1. The K value is set to 30 and W^R is set to 1.

Table 4.1: Simulation parameters for service class

Traffic Class	Max Flows	Rate [Kbps]	C_{OUT} [Mbps]
EF	10	400	0.6-3.4
AF	10	800	0.6-3.4
BE	10	3400	0.6-3.4

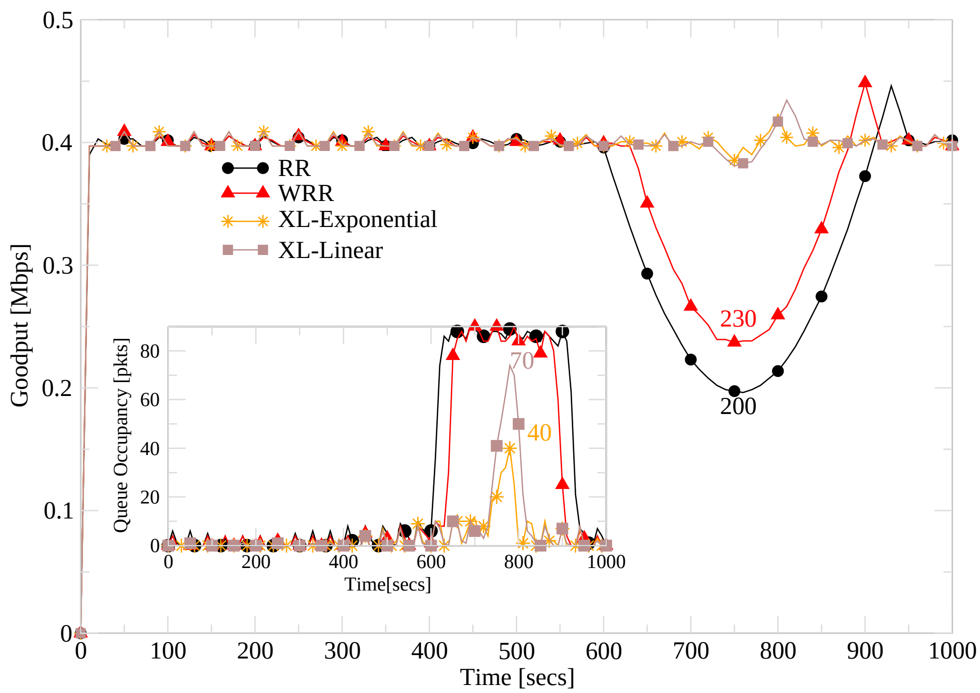


Figure 4.4: EF Goodput performance results and average queue occupancy when using RR, static WRR, XL-linear and XL-exponential algorithms

The simulation results of goodput and queue occupancy for the EF traffic class are shown on Fig. 4.4. Here the four mechanisms working independently are compared, these are: RR, static WRR, XL-linear and XL-exponential.

As it is observed on Fig. 4.4, using the RR algorithm (symbol •) the EF traffic class is not guaranteed when a reduction of bandwidth is experienced. Particularly, at the 750-seconds time point (see the valley) only 200 Kbps of goodput level is reached. Similarly, when using the WRR mechanism with static weight values (symbol △), the EF traffic class is able to reach 230 Kbps of goodput at the same time point. Although this goodput level is enhanced compared to the previous case, this is not enough to guarantee the high priority traffic class which is set to 400 Kbps. In both cases, the queue occupancy is overloaded, reaching its limit value (set to 90 packets), leading to an increase of the system latency.

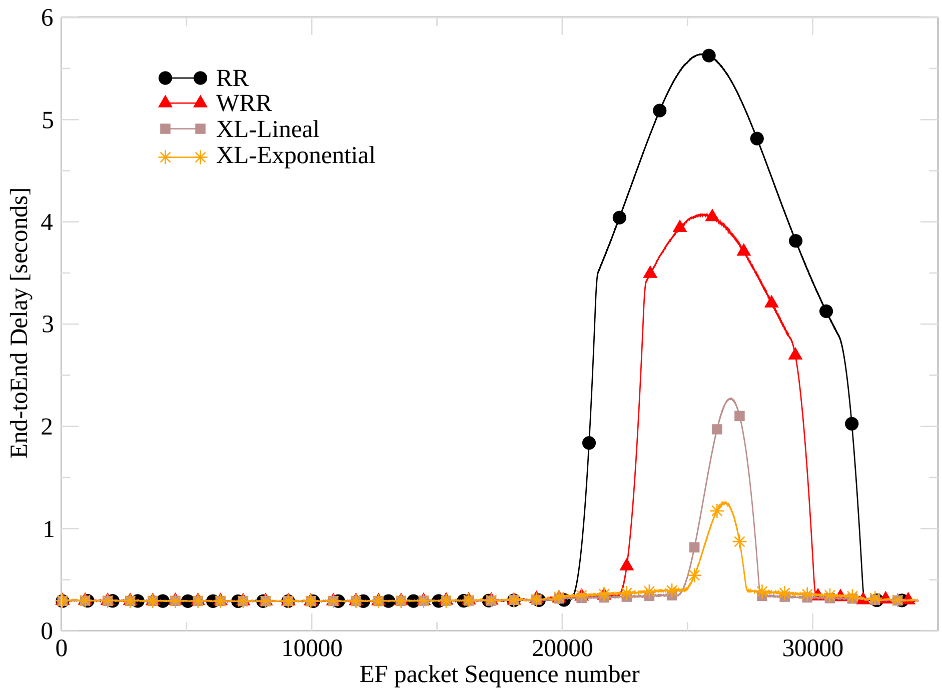


Figure 4.5: EF per packet end-to-end delay

In contrast to these results, when using the proposed IP scheduler (either XL-linear or XL-exponential), the EF goodput is able to reach its 400 Kbps during all the simulation (see symbols ■ and symbol ∗).

In both cases, the EF buffer occupancy is reduced compared with the RR and static WRR. Particularly, when the linear function is employed (XL-linear), a faster variation of the cross-layer values is generated and thus a peak value of 70-packets buffer occupancy is reached. However, when the exponential IP scheduler (XL-exponential) is used, the buffer occupancy for the EF traffic class is kept at lower levels, being able to reach 40 packets

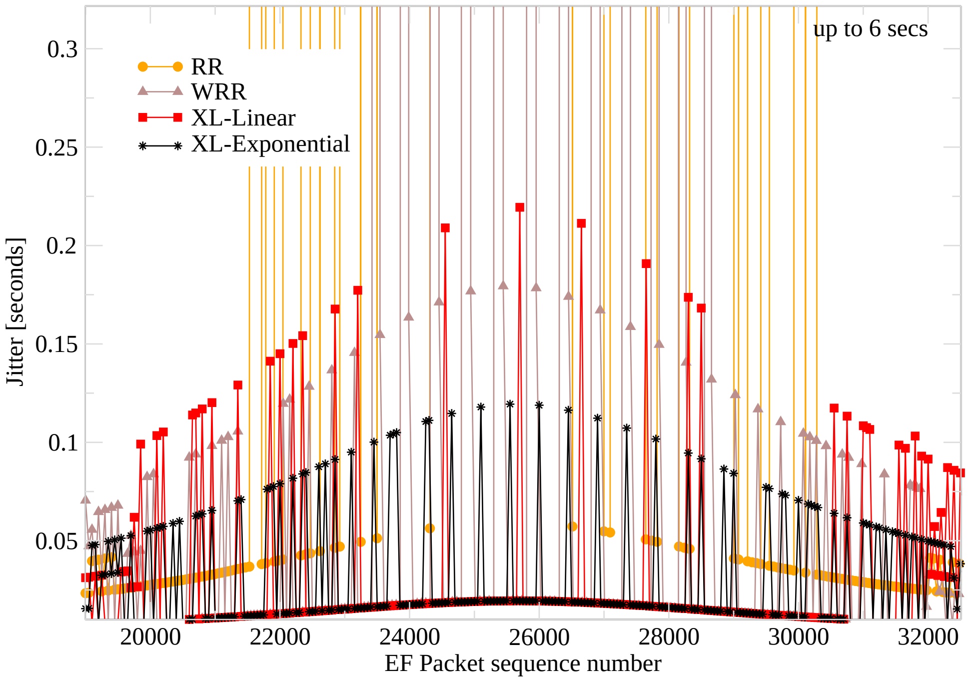


Figure 4.6: EF per packet jitter

even though an extreme reduction of bandwidth is experienced. This is a suitable result when working with QoS DVB-S2 satellite systems.

In order to have a perspective related to the delay, Fig. 4.5 shows the one way end-to-end delay experienced by each EF packet (considering an example of one EF flow). As it is observed, in the interval starting from 20,000 to 30,000 packet sequence numbers (which is the interval when the bandwidth capacity is reduced), the peak delay, using the RR mechanism, reaches values between 1 to 5.5 seconds. This situation causes longer delays at the application layer. Similarly, when using static WRR the delay values range between 1 to 5 seconds. This is mainly because, neither the RR nor the WRR algorithms are able to guarantee a minimum delay value when the satellite experiences a capacity reduction.

Nevertheless, when either the linear or the exponential algorithm are activated, the end-to-end delay values are considerably reduced, having a peak, at the *Scarce Capacity interval*, that reaches 2 seconds for the XL-Linear and 1 second for the XL-Exponential (see Fig. 4.5). In addition, Table 4.2 shows the mean and the standard deviation (Std. Dev.) parameters for the EF delay experienced at the application layer. As it is observed, both values are highly reduced when using the proposed XL-Scheduler (either linear or exponential) compared to those results obtained by using either RR or WRR algorithms.

In particular, using the XL-Exponential scheduler it is possible to guarantee mean delay values up to 330 msec (see Table 4.2) which are quite bigger than the one-way propagation delay (set to 280 msec). This is a suitable result considering the fact that the available capacity present in the satellite systems is reduced to a critical point in which the priority of the EF traffic class requires to be guaranteed.

Table 4.2: E2E delay parameters for the EF traffic class

Parameter	RR	WRR	XL-linear	XL-exponential
Mean delay (seconds)	3.71	2.45	0.37	**0.33**
Std. Dev. (seconds)	1.7	1.56	0.59	**0.24**

Finally, Fig. 4.6 shows the jitter values experienced by each EF packet at the application layer. Specifically, these results are measured during the *Scarce Capacity interval*. Here, it is possible to observe that when using either RR or WRR, the peak jitter values are up to 6 seconds. Nevertheless, when using the proposed cross-layer scheduler the same peak values are between 0.2 and 0.1 seconds. Table 5.1 shows the mean and the standard deviation (Std. Dev.) parameters for the EF jitter. As it is observed, when either the XL-Linear or the XL-Exponential schedulers are activated the mean jitter values are up to 16 msec, which are suitable values to guarantee the QoS specification for the EF traffic class [JNP99].

Table 4.3: Jitter parameters for the EF traffic class

Parameter	RR	WRR	XL-linear	XL-exponential
Mean jitter (seconds)	6.55	0.01740	0.01666	**0.01624**
Std. Dev. (seconds)	3.81	1.93	0.0435	**0.02921**

It is worth mentioning that when performing a detailed analysis considering the proposed XL-scheduler mechanism working simultaneously with different number of traffic flows (i.e. 5 flows per each traffic class) and also different EF traffic loads, similar enhancements have been obtained. However, in this subsection, the most illustrative case is described, representing a situation in which the satellite capacity is extremely reduced by the presence of a heavy rain event.

4.3.1 QoSatArt: RQM performance evaluation

This subsection is defined in order to evaluate the RQM mechanism proposed in Section 3.4.2, in comparison to the drop tail scheme. Here, the AQM system showed on Fig.3.3 has been simulated using the NS-2 simulation tool. Particularly, in this test the bottleneck satellite link is set to 2 Mbps, a standard Sack TCP is considered and the transmission rate for the EF and the AF traffic classes is set to 200 Kbps and 1 Mbps respectively.

The simulated scenario evaluates the performance of the EF and AF traffic classes under different traffic conditions. Specifically, different number of flows for the EF and

AF traffic classes are transported (either eight or twelve flows). The aim of this test is to overload the high priority traffic classes in order to activate the RQM mechanism, sending the highest number of out-of-profile packets to the BE queue.

As a result, the BE queue will become flooded with out-of-profile packets leading to the activation of the proposed re-queuing mechanism. The goal of this simulation test is to evaluate the performance of high priority traffic classes in terms of goodput, delay and jitter, and compare the results with those achieved by using the drop tail mechanism.

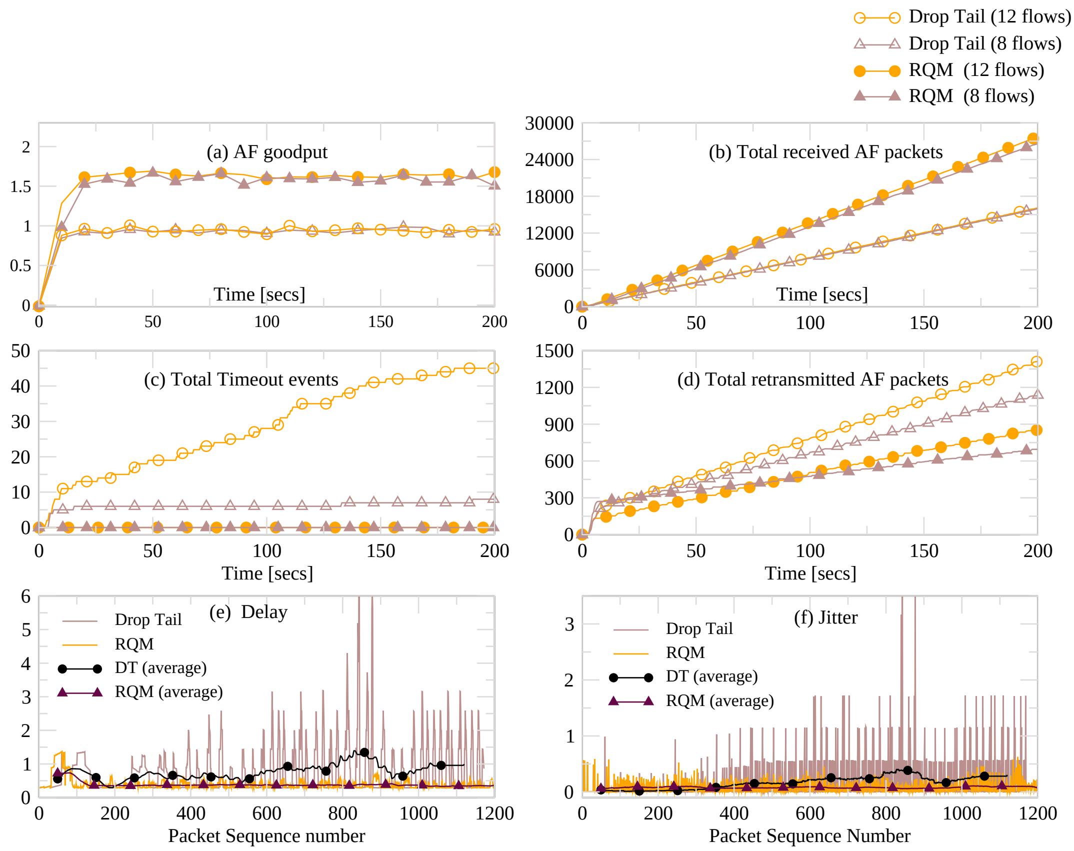


Figure 4.7: Simulation results for the AF traffic class (eight and twelve flows) using either the proposed RQM or the drop tail mechanism. (a) AF goodput, (b) Total received AF packets, (c) Timeouts number, (d)Total retransmitted AF packets, (e) Delay and (f) Jitter

Fig. 4.7 shows the simulation results for the AF traffic class using both queuing options: either the RQM or the drop tail mechanism. These results consider the average values calculated when either eight or twelve AF flows are working simultaneously. Particularly, Fig. 4.7.b illustrates the total received AF packets at the application layer. As it is shown, when using the proposed RQM mechanism (symbols • and △), more packets (approx. 27,000) are correctly received in both cases (eight or twelve flows) during all the simulation time. In contrast to the drop tail scheme in which the levels of received packets (symbols •, △) are less (approx. 15,000). Therefore using RQM, the number of correctly received packets have increased in 44% (on average) compared with the drop tail scheme. This result is caused mainly because when using the proposed RQM mechanism, the out-of-profile AF packets are not dropped. Instead, these packets are sent to the BE queue with a downgraded QoS level, giving them the chance to reach the receiver before a RTT.

As a natural consequence, the more AF packets are received, the more the goodput level is enhanced. This result is shown on Fig. 4.7.a in which it is possible to observe that using the RQM mechanism, the goodput level is enhanced reaching 1.5 Mbps. In contrast to the drop tail mechanism that drops all the out-of-profile AF packets, reaching only 1 Mbps of goodput. This result represents a significant goodput enhancement of 33% (on average) when using RQM.

This means that the RQM mechanism enables the TCP traffic (in this case, the AF traffic class) to take advantage of the resources that are not used (i.e. BE class).

In a similar way, Fig. 4.7.d. shows the number of retransmitted packets. As it is observed, when the proposed RQM mechanism is used, the number of retransmitted packets is lower compared to the drop tail scheme at every time it is measured. Similarly, Fig. 4.7.c shows the number of timeout events present during the simulation. As it is depicted, no timeout events are present when the proposed RQM mechanism is used, which is a desirable situation. In contrast to the drop tail scheme which suffers from many timeout events during this simulation.

In this context, it becomes important to have a perspective related to the delay. Therefore, Fig. 4.7.e shows the one-way end-to-end delay experienced by each AF packet (considering an example of one flow). As it is observed, when using the drop tail scheme, the peak delay values range between 2.5 and 6.1 seconds, causing longer delays at the application layer. This is done, given that out-of-profile AF packets are just discarded.

In this way, the sender uses three consecutive dupACKs as a signal for retransmitting the missing segment (fast retransmit). After that, the fast recovery algorithm is used until the first non-duplicated acknowledge is received.

Nevertheless, when the RQM mechanism is activated, the end-to-end delay values have peak points that reach less than 0.5 seconds, (see Fig. 4.7.e), which are clearly lower than the RTT_{min} (set to 560 msec). Here, the peak values are mainly caused as a result of the packet disorder experienced at the application layer.

To see the effects of packet reordering in triggering the fast retransmit/fast recovery algorithms, it is important to keep in mind that by using the RQM mechanism, the out-of-profile AF packets are not discarded but these are instead downgraded by being re-queued to the BE queue. Obviously, this re-queuing stage might also cause gaps in the sequence of data segments, which will cause the receiver to send dupACKs for each out-of-order received segment. As usual, the sender will use fast retransmit to send (what appears to be) a missing segment upon the arrival of three dupACKs.

With the drop tail scheme, the period of time when the receiver detects the gap until it obtains the retransmitted packet, it is always greater than the RTT_{min}. Conversely, the improvement achieved by using the RQM is that it requires less time to deliver the out-of-profile AF packets to the receiver, since these packets, in fact, have not been discarded.

Nevertheless, in the RQM mechanism, this period of time depends on the time the re-queued segment stays in the BE queue, which is related to the magnitude of the disorder (the amount of segments from the same flow that are sent through the channel between the gap and the re-queued segment). Notice that in the RQM mechanism, this time is always equal or lower (and can be actually much lower) than the time required by the drop tail scheme (see Fig. 4.7.e). The final result is that the application layer at the receiver get the data earlier, which improves the QoS experienced by the end user.

Similarly, Fig. 4.7.f shows the jitter values experienced by each AF packet at the application layer. As it is observed, when the RQM mechanism is activated the peak jitter has values ranging 0.4 seconds compared to the drop tail scheme which has peak values around 1.2 seconds.

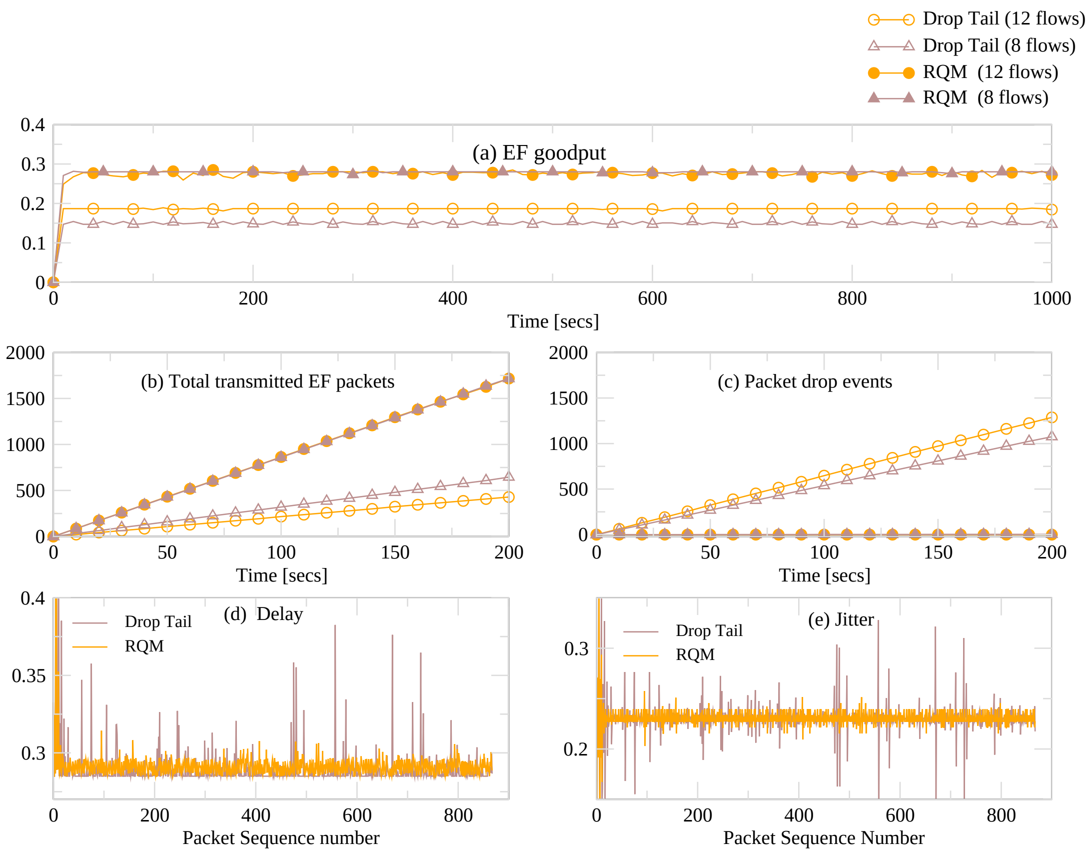


Figure 4.8: Simulation results for the EF traffic class (eight and twelve flows) using either the proposed RQM or the drop tail mechanism. (a) AF goodput, (B) Total transmitted EF packets, (c) Number of packet drop events, (d) Delay and (e) Jitter

Finally, Fig. 4.8 shows similar simulation results for the EF traffic class using both queuing options: either the RQM or the drop tail mechanism. Here, the same performance parameters have been evaluated such as goodput, total transmitted packets, number of drop events, delay and jitter values. As it is observed, when using the proposed RQM to transport the EF traffic class (using UDP protocol), similar results have been obtained compared to the AF traffic class.

With the obtained results it is possible to conclude that when the RQM mechanism is employed, the EF and AF goodput levels are improved, while the end-to-end delay and the jitter experienced by the user application are reduced. As a consequence, the Quality of Experience (QoE) experienced by end-users is enhanced.

It is worth mentioning that a detailed analysis has been conducted that consider the proposed RQM mechanism working simultaneously with different number of traffic flows and load conditions. Such analysis has turned out to get similar enhancements. Here, the most illustrative cases have been described.

4.3.2 QoSatArt with BW variations: performance results

In this subsection, the complete QoSatArt design (defined in Section 3.3) is evaluated including the AQM system, the cross-layer IP scheduler and the RQM mechanism considering the typical GEO satellite characteristics in the presence of bandwidth variations. The objective of this performance evaluation is to test the QoSatArt architecture considering extremely traffic load conditions affecting the satellite link due to a heavy rain event.

In particular, the DVB-S2 system response when adopting the QoSatArt architecture is evaluated and compared these results against two proposals. The first comparison includes the Round Robin mechanism as a packet scheduler together with a simple drop tail queue as a discarding mechanism. This is called a *RR-basic* configuration. Similarly, the second comparison considers the same drop tail queuing mechanism together with the Weighted Round Robin scheduler that uses static weight values set to μ_{EF}, μ_{AF} and μ_{BE} for the EF, AF and BE traffic classes respectively. These values are set considering the proportional distribution of resources [DR99]. This configuration is called a *WRR-static*. Finally, a third comparison is performed considering our proposed QoSatArt design including the AQM system, the cross-layer IP scheduler and the RQM mechanism. As it is designed QoSatArt should enforce the priority levels taking into account the link capacity variations reported by the physical layer using the proposed cross-layer optimization. Therefore, the QoSatArt should be able to guarantee high priority traffic classes regardless of the weather and traffic load conditions.

To perform this comparison the goodput and queue occupancy levels for the EF, AF and BE traffic classes are evaluated when a heavy rain event reduces the bandwidth availability. Here, the bandwidth variability is represented by a sinusoidal wave period having 1000-seconds time duration. For such wave, the interval between 0 and 500 seconds represents a situation in which the satellite system experiences clear sky conditions. Conversely, after 500 seconds the capacity starts reducing because of the presence of a heavy rain event. Here, the satellite capacity reaches a minimum value set to 400 Kbps

Table 4.4: Performance Evaluation Parameters

Traffic Class	**Max Flows**	Rate [Kbps]	BW variations [Mbps]
Test 1		μ_{EF} variations	
EF	5	100-500	
AF	10	800	3.4 - 0.4
BE	10	2300-2700	

at 750-seconds time point. After this point, the capacity starts increasing up to reach its steady-state condition.

(i) Test 1: μ_{EF} variations,the incoming EF rate suffers from variations.

Table 1 shows the performance evaluation parameters used to conduct each test, including the number of flows, the incoming rates and bandwidth fluctuations caused by a heavy rain event.

The objective of this test is to evaluate the priority levels that each traffic class is able to get when the EF traffic rate experiences bandwidth fluctuations between 100 to 500 Kbps.

Fig. 4.9 shows the goodput and queue occupancy for the EF, AF and BE traffic classes working simultaneously considering only the QoSatArt architecture in the presence of a heavy rain event. Here, it is possible to see that the EF traffic class (symbol $\bullet$) experiences bandwidth variations represented by a simple step function ranging between 100 to 500 Kbps. As it is observed, the variable goodput for the EF traffic class is totally guaranteed even though a extreme reduction of bandwidth capacity is present in the satellite system.

Moreover, using the QoSatArt it is possible to assign all the available resources to the EF traffic class when a reduction of capacity is experienced (see interval between 600 and 800 seconds). Nevertheless, at the 750-seconds time point, the AF traffic class (see symbol is able to have more bandwidth resources than the EF traffic class, given that, at this point, the EF traffic rate is reduced. Therefore, the AF traffic class takes advantage of the resources that the EF traffic class does not use.

In particular, adopting the QoSatArt when EF rate variations are experienced, it is possible to provide the highest priority for the EF traffic class, while the AF traffic class has more priority than the BE class. This is an expected result according to the *QoS policy* defined for a DVB-S2 satellite system. As it is observed, The BE traffic class (symbol $*$) is able to use the remaining link bandwidth during all the simulation, taking advantage of the resources that the other classes are not using.

Finally, Figs. 4.10 and 4.11 show the results considering the *RR-basic* and *WRR-static* respectively, for the EF, AF and BE traffic classes working simultaneously. In both graphs it is possible to see that neither the *RR-basic* nor the *WRR-static* are able to provide QoS guarantees for high priority traffic classes when EF traffic variations and bandwidth

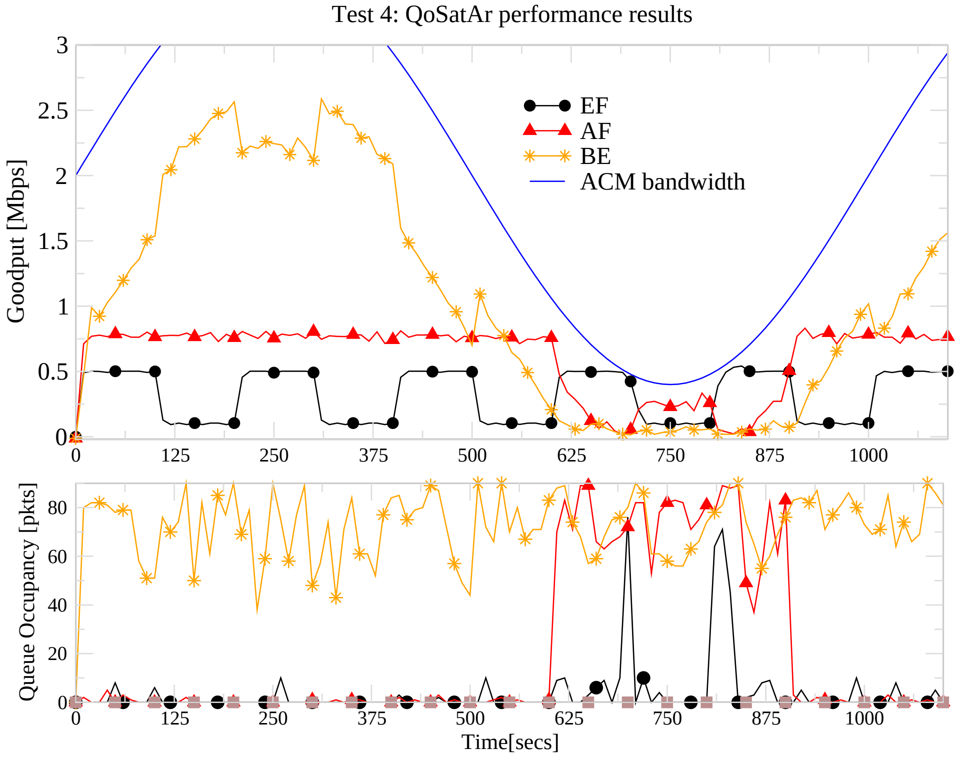


Figure 4.9: Test 1: QoSatArt simulation results in the presence of EF variations

reductions are experienced. Similarly the queue occupancy levels are overloaded (see interval between 600 and 800 seconds) when a bandwidth reduction is faced due to a heavy rain event. Here, it is possible to see that the EF buffer occupancy level is reduced when adopting the QoSatArt architecture compared to the cases when using either the *RR-basic* or the *WRR-static*.

4.4 Conclusions

In this chapter, the detailed design of an IP scheduler based on a cross-layer mechanism has been presented. It takes into account the link bandwidth variations present in the DVB-S2 forward link to provide QoS guarantees for different traffic classes. Based on this QoS prioritization, it is possible to determine the minimum amount of resources for high priority traffic classes.

The IP scheduler design includes a cross-layer mechanism between the physical layer and the network layer to provide QoS guarantees while enhancing the bandwidth allocation among traffic flows requiring QoS.

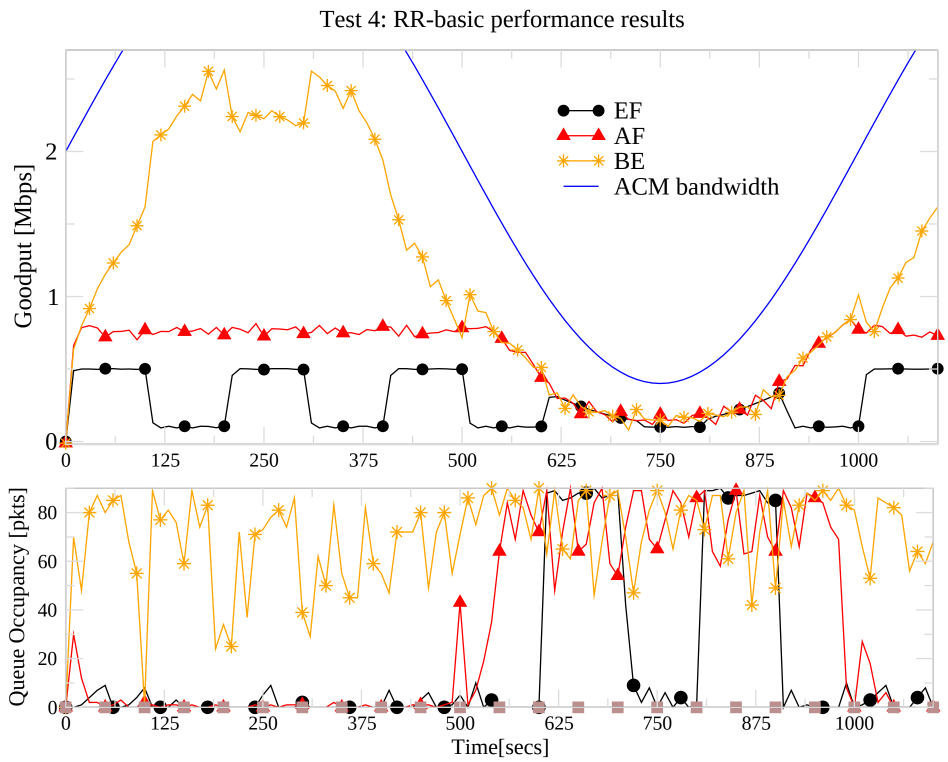


Figure 4.10: Test 1: RR-basic simulation results in the presence of EF variations

The available bandwidth is determined considering the ACM adaptation behaviour reported by each RCST terminal. Here, a characterization of the link bandwidth variations has been performed using a sinusoidal approximation to determine the intensity of the rain event affecting the bandwidth availability.

The algorithm for calculating the weighted values to provide QoS guarantees is a modification based on the PDS model which includes an exponential function to associate the QoS requirements for each traffic class with the satellite link capacity.

Additionally, the cross-layer information is interchanged among agents using a signaling mechanism based on SNMP protocol.

By employing the NS-2 simulator tool the performance of the cross-layer IP scheduler has been studied and compared against the RR and WRR mechanisms. The proposed scheduler is able to ensure the QoS requirements for each traffic class while enhancing the bandwidth assignment for flows requiring QoS guarantees when bandwidth reductions caused by rain events are experienced. The simulation results have shown that the exponential IP scheduler enforces the priority levels and guarantees the QoS parameters established in the SLA, while maintaining the queue occupancy, delay and jitter parameters at lower levels, leading to a reduction of the system latency.

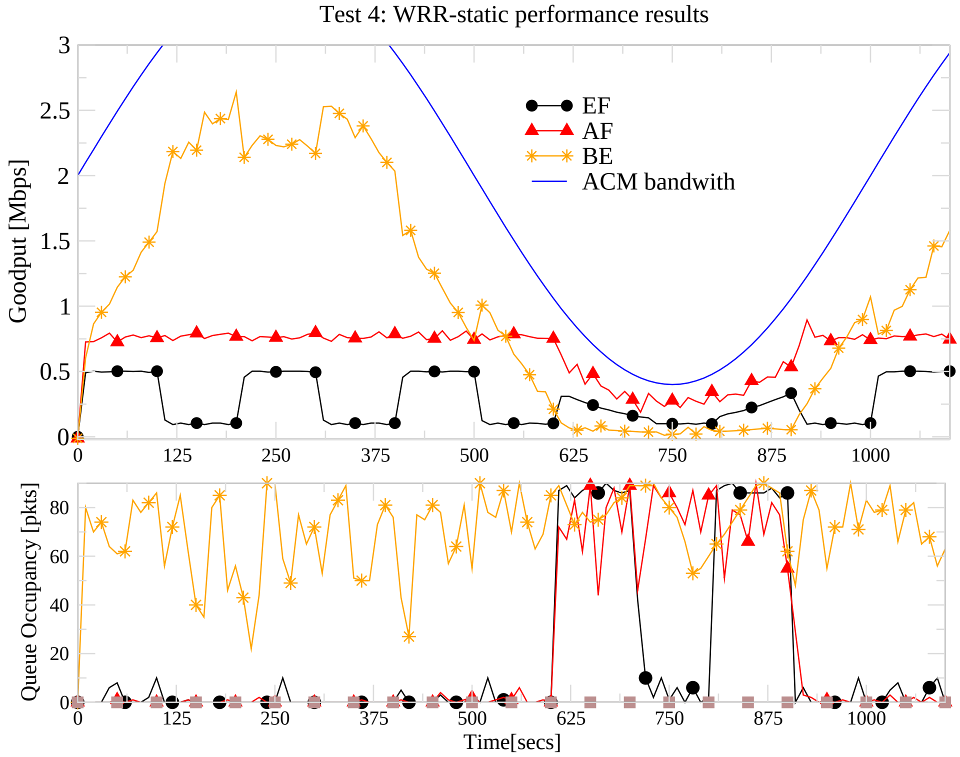


Figure 4.11: Test 1: WRR-static simulation results in the presence of EF variations

The key results obtained when evaluating QoSatArt architecture show that it is possible to keep control of the satellite system load while guaranteeing QoS levels for the high priority traffic classes even though bandwidth variations due to rain events are experienced. The simulation results also demonstrate that with the adoption of the proposed RQM mechanism, the user's QoE is improved while keeping bounded the delay and jitter values for high priority traffic classes. In particular an AF goodput enhancement of 33% (on average) is reached.

Moreover, with the evaluation of the cross-layer IP scheduler, the high priority traffic class is always guaranteed regardless of the channel conditions affected by rain events. The most important aspect in QoSatArt it is that the proposed architecture is able to guarantee the QoS requirements for specific traffic flows using a single parameter: the bandwidth availability. This parameter is set at the physical layer (considering ACM adaptation) and sent to the IP scheduler taking advantage of the proposed cross-layer optimization.

CHAPTER 5

QoSatArt: Analysis of TCP variants

In the previous chapters, the attention has been payed on the design of the QoSatArt architecture to ensure QoS requirements focused on the SI layers. Specifically, the definition of algorithms and a signaling mechanism working basically on the network layer have been proposed. However, to efficiently address QoS issues across the upper layers, it is important to consider the intrinsic characteristics present in satellite systems that strongly affect the performance of protocols working at the higher layers.

In particular, applications such as file transfers, HTTP web browsing and ssh applications rely on the TCP protocol, which is strongly affected by the GEO satellite characteristics such as the large propagation delay and the bandwidth variations in DVB-S2.

The two-way propagation delay, also called minimum Round Trip Time (RTT_{min}) affects the TCP performance because the congestion control of this protocol takes up several seconds before reacting to/or recovering from congestion events. The TCP congestion window (*cwnd*) size is determined by the successful acknowledgement reception per RTT [LM97], [PFTK98], [MSMO97]. Thus, the longer the RTT, the narrower the *cwnd* growth and the slower TCP response is experienced.

Considering such TCP limitations, it is clearly observed that even if the lower layers techniques are able to offer guaranteed bandwidth to TCP flows, the TCP protocol itself is not be able to make effective use of the network resources as a result of its intrinsic design. To address this the implementation of a transport layer able to adapt to the environmental characteristics in order to make more effective the provision of QoS guarantees is required.

In this chapter, the possibility to directly adopt a selected TCP variant on an end-to-end (E2E) environment (trying to keep the E2E semantic) with QoS provisioning has been studied. The proposed environment is the QoSatArt architecture (described in Section 3.3.1) which has been designed based on the ETSI-BSM-QoS framework to provide QoS guarantees. It includes the main functional blocks such as *the AQM system, the RQM mechanism, the XL-Manager and the IP scheduler* showed in Fig. 3.2 and Fig. 3.3. Here, the performance evaluation of several TCP variants is carried out in order to choose the more suitable TCP variant that enhances the performance of the proposed QoSatArt architecture. In the view of the author, this is the first time that such analysis is performed considering an ETSI-BSM-QoS scenario.

The analysis of TCP variants considers the fact that there are many different official TCP variants. However, most of these variants have proven to have poor performance when working over satellite systems. This is mainly because the satellite path delay not only produces a long connection set up time but also produces low bandwidth utilization. This last issue is the result of the slow-start mechanism and the limited TCP window size, which can be increased but by default it is usually below the optimal size for broadband satellite networks.

The proposed analysis of TCP variants can be seen as a continuation of the work developed by Caini et al. [CFL+09], in which several TCP variants were analyzed over a DVB-S GEO satellite test-bed. In that work, the authors studied the performance of several TCP variants considering the DVB-S forward link where QoS was not considered. The results of such work showed that the TCP variants Hybla and BIC achieved the best performance in terms of goodput. Here, these outcomes have been taken as a starting point to develop the research work presented in this chapter.

In fact on the present analysis, the BIC TCP variant has been replaced by its enhanced version called CUBIC TCP. Moreover, the Sack TCP variant has been added representing a baseline TCP variant.

Hence, CUBIC TCP, Hybla TCP and Sack TCP are the selected TCP variants to carry out the performance evaluation. The metrics for analyzing the behavior of the proposed TCP variants are the goodput, queue occupancy (related with delay) and TCP friendliness. The performance evaluation is carry out considering the *fairness* and *friendliness* analysis.

5.1 QoSatArt satellite scenario

In order to perform the analysis of TCP variant, the satellite scenario for the QoSatArt architecture which is shown in Fig. 3.1 has been simulated using the NS-2 simulation tool. It includes the main functional blocks (such as *AQM, RQM, XL-Manager and IP scheduler*) which has been added and programed using TCL code.

The settings used to conduct the analysis are: the satellite channel capacity considered as the bottleneck link is set to 2 Mbps. The minimum Round Trip Time (RTT_{min}) considered as the two-way propagation delay value is set to 560 msec The buffer length of each traffic class is set to the BDP value that yields 90 packets. The forward link is based on the DVB-S2 standard which uses the ACM scheme to achieve lower Packet Error Rate values. To perform the simulation, it is considered a Bit-Error-Rate value set to BER = 10^{-7}. The Committed Information Rate (CIR) for the EF and AF token buckets are set to 0.4 Mbps and 0.8 Mbps respectively. The MTU size is set to 1500 bytes. The DVB-RCS link has permanent bandwidth capacity set to 256 Kbps. It is also assumed that there are not any lost ACK messages and that ACK messages have more priority level than data packets when they are sent through the RCS terminal.

The SLA example is defined to enforce the TB rate limiters considering the bottleneck satellite link set to 2 Mbps. In this condition, the nominal transmission rate for each traffic class are set to 400 Kbps for the EF traffic class and 800 Kbps for the AF traffic class, while the BE traffic class will use the remaining bandwidth. Particularly, the AF and BE traffic classes are transported using one of the following TCP variants: Sack TCP, Hybla TCP and CUBIC TCP.

It is worth mentioning that the set of simulations were carried out considering an aggregated flow per traffic class. This was done to simplify the analysis in order to obtain comparable figures between the AF and BE traffic classes. Additionally, the EF traffic class was not included in the graphs presented in this set of experiments, as it has a constant behavior which is not relevant for the TCP analysis performed in this section. However, although these results are not shown in the graphs, they were considered in the simulations.

Here, the following TCP analysis have been performed:

(i) *Fairness*: it is defined to analyze the congestion window evolution (*cwnd*) and goodput response when the proposed QoSatArt architecture (including its main functional blocks) is used, in comparison with a scenario in which **no** QoS guarantees are supported.

(ii) *Friendliness*: it is defined to analyze the goodput and the buffer occupancy levels when different TCP variants are used to transport the AF and BE traffic classes, considering the QoSatArt scenario.

(iii) *Traffic and Bandwidth fluctuations*: the analysis of TCP variants considering EF traffic variation and bandwidth fluctuations are evaluated.

5.1.1 Performance metrics and the selected TCP variants

To carry out the performance evaluation of the selected TCP variants the metrics already defined in Section 4.2.3, such as *goodput*, *queue occupancy*, *one-way delay* and *jitter*, are considered. However, the following metrics are also defined for evaluating the TCP-friendliness and fairness level:

(i) The *TCP-friendliness* level represents a generic term that describes the ability of a TCP variant to coexist with *another* TCP variant and fairly share the available bandwidth in the same bottleneck link [TLL07]. Particularly, in our scenario the friendliness level is measured between the TCP variants used to transport the AF and BE traffic classes. In this context, an enhanced friendliness level is considered if a TCP variant (used to transport the AF traffic class) is able to evenly distribute the available bandwidth with a second TCP variant (used to transport the BE traffic class). As a result, each TCP variant must be able to obtain around 800 Kbps of goodput, considering that the available bandwidth for both classes (AF and BE) is set to 1.6 Mbps.

(ii) Similarly, the *TCP-fairness* level represents a generic term that describes the ability of a TCP variant to coexist with the *same* TCP variant and fairly share the available bandwidth in the same bottleneck link.

On the other hand, the technical issues related to the selected TCP variants are described as follows.

(i) Sack TCP [BAFW03] is likely the most widely adopted variant on wired networks. It follows the guidelines of the Van Jacobson congestion control mechanisms, where the Selective Acknowledge (SACK) option was introduced to deal with multiple packet losses occurring at the same data window. By adding the SACK mechanism, the sender is able to have information about the data segments that have been received allowing only the retransmission of the missing segments. This enables the sender to recover more than one packet per RTT.

(ii) CUBIC TCP [HRX05] is an enhanced version of its predecessor BIC-TCP (Binary Increase Congestion control). It uses a cubic function to control the congestion window growth. To improve the TCP scalability over fast and long distance networks, CUBIC TCP defines a maximum congestion window (W_{max}). This value is used as a reference when a loss event occurs. Thus, when a window reduction is experienced, if the *cwnd* value is far from W_{max}, the *cwnd* growth will be fast. On the contrary, if the *cwnd* value is close to W_{max}, its growth will be slow. The congestion window size is set to $cwnd = C(T-K)^3 + W_{max}$, where T is the elapsed time since the last loss event has occurred and $K = (W_{max}\beta/C)^{1/3}$, where β and C are set to 0.2 and 0.4 respectively. The key feature of CUBIC TCP is that the window growth only depends on the time between two consecutive congestion events. This allows CUBIC TCP to be independent of the RTT and to fairly share the available bandwidth with other TCP variants and other TCP connections established along short and long paths, i.e. small and big $RTTs$.

(iii) Hybla TCP [CLF+06] also tries to obtain the same instantaneous transmission rate $B(t)$, for connections with small or big $RTTs$. To achieve this, Hybla TCP replaces the standard congestion control by two equations: $W_{i+1} = W_i + 2^\rho - 1$ for *slow start phase* and $W_{i+1} = W_i + \rho^2 \,/\, W_i$ for *congestion avoidance phase*. Where, ρ is defined as a normalized round trip time ($\rho = RTT \,/\, RTT_0$). The minimum value for ρ is set to 1, to avoid slowing down the connections with $RTT < RTT_0$. By implementing these modifications, it is possible to achieve independence between the instantaneous transmission rate $B(t)$ and the actual RTT value.

5.2 Fairness Analysis

In this section the fairness analysis consists on evaluating the congestion window (*cwnd)* evolution between two TCP flows (the AF and BE traffic classes) using the same TCP variant.

The results when the proposed QoSatArt architecture is used are compared in contrast to those results achieved when **no** QoS guarantees are performed. As an essential requirement, the selected TCP protocol should guarantee a fair behavior among traffic flows using the same TCP variant.

Therefore, this simulation test is proposed to analyze the intra-protocol behavior in order to evaluate its impact in the satellite network when the all the sources use the same TCP variant.

Fig. 5.1.a shows the simulation results when the QoS guarantees are not considered. In contrast, Fig. 5.1.b shows the simulation results when the proposed QoSatArt architecture is activated. The goodput results for each case are shown in Fig 5.1.c.

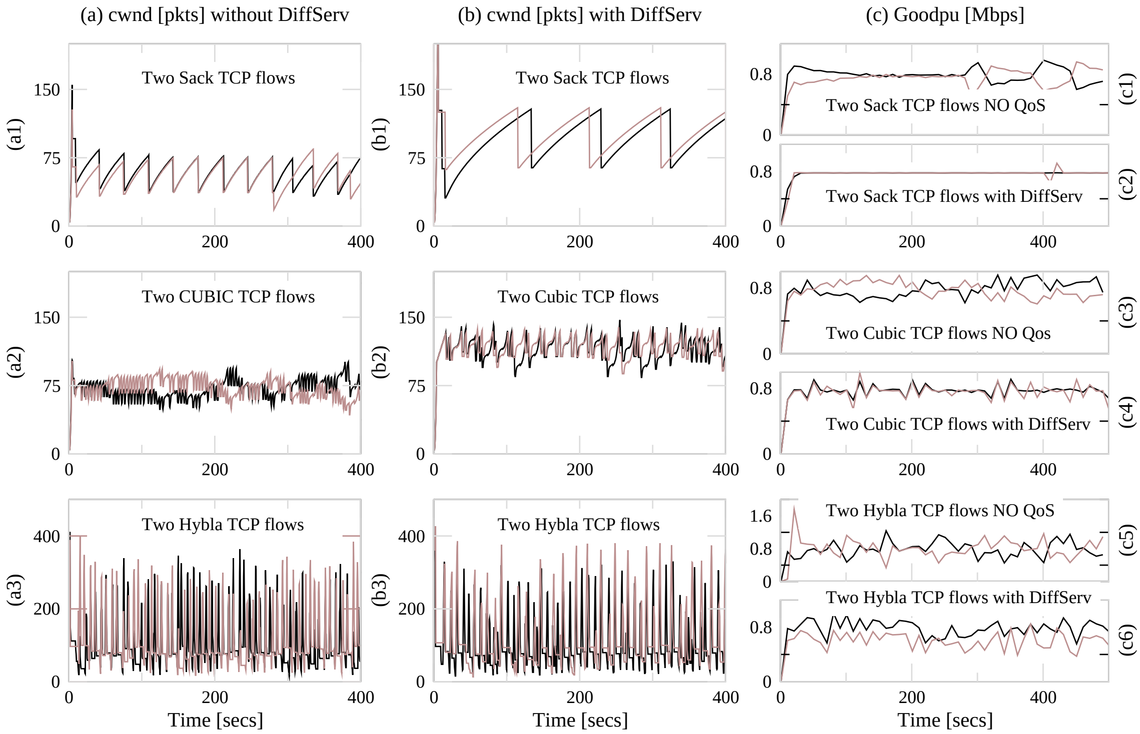


Figure 5.1: Congestion window evolution and Goodput results: (a) without QoS guarantees, (b) with the QoSatArt architecture and (c) the corresponding goodput results.

As it is observed for the case of Sack TCP, depicted on Fig. 5.1.a1 and 5.1.b1, when the same TCP variant is used the typical Reno TCP saw-tooth behavior is shown, either when QoS guarantees are implemented or not, as a response of applying the standard Additive Increase Multiplicative Decrease (AIMD) algorithm. In contrast with the previous case when no QoS guarantees are supported both *cwnds* grow faster reaching peak values within the range of 75 packets. However, when the QoSatArt architecture is activated, both *cwnds* grow slower compared to the previous case, reaching peak values up to 110 packets. This situation is reflected in the goodput levels reached by each flow and illustrated on Figs. 5.1.c1 and 5.1.c2. For the case when no QoS guarantees are supported the goodput level that both flows are able to reach exhibit an unfair distribution of resources in some specific time points (see Fig. 5.1.c1). However, when the proposed QoSatArt architecture is activated (see Fig. 5.1.b1), both flows essentially waste the same amount of bandwidth (800 Kbps) enhancing its fairness level.

In a similar way, Figs. 5.1.a2 and 5.1.b2 show the *cwnds* evolution when two CUBIC TCP flows are used. In both cases, it is possible to note that the curves have plateaus around the maximum *cwnd* values corresponding to the window sizes at the time when the last packet loss event occurred. In the case where no QoS guarantees are supported both flows cause disturbance within competing TCP flows, reaching peak values within the range of 75 packets on average. Nevertheless, when the QoSatArt architecture is activated, both *cwnds* curves are smooth and do not cause much disturbance to competing TCP flows. The convergence of both flows to a fairly sharing state occurs at the beginning of the simulation, reaching peak values up to 150 packets.

On Figs. 5.1.c3 and 5.1.c4 it is possible to note that the *cwnd* evolution directly impact the goodput levels. In the first case both flows exhibit oscillating goodput levels, being unable to fairly share the available bandwidth. Here, the competition for resources is also evident as in some time periods, the goodput for the priority class is not guaranteed. In contrast to this, on Fig. 5.1.c4 it is possible to see that both flows compete for the available resources, being able to fairly share the remaining bandwidth. In addition the AF transmission rate is always guaranteed over the BE traffic class.

Finally, Figs. 5.1.a3 and 5.1.b3 show the *cwnd* evolution within the interaction of two Hybla TCP flows. In both graphs, the *cwnds* show consistent and large oscillations. This is mainly because Hybla TCP sets its *cwnd* to the defined maximum size at the beginning of the simulation, trying to fill a great amount of packets into the link. As a result, its window size shrinks considerably having a high oscillation. The oscillations are mainly due to increase in RTT which are generated by the detection of packet losses or congestion events. This situation increases the growth of ρ (where $\rho = RTT \ / \ RTT_0$ for Hybla TCP), which in turn leads to even a larger *cwnd*, filling the network with packets. These oscillations persisted even in very long simulation runs, and also when the QoSatArt architecture was activated.

The goodput level is shown on Figs. 5.1.c5 and 5.1.c6. When QoS guarantees are not considered, both flows are unable to fairly share the available bandwidth, exhibiting an oscillating and unfair behavior. Similarly, when the proposed QoSatArt architecture is implemented (see Fig. 5.1.c6), the oscillating behavior persists, however the goodput level for the AF traffic class keeps its priority level during all the simulation time.

As a general conclusion of this set of simulations we have validated that either using the DiffServ to provide QoS guarantees or not, the *cwnd* evolution works according to the congestion control algorithm defined for each TCP variant. In addition, when the proposed QoSatArt architecture is used, the AF traffic class keeps its priority level over the BE traffic class, while the *cwnds* evolution (in the cases of Sack TCP and CUBIC TCP) evolve without any interference between each other. For the case of Hybla TCP, its aggressive behavior was observed, as it tries to put a great amount of packets in the satellite link, overloading the queues and therefore affecting its level of goodput. This behavior will be further analyzed in the following experiments.

5.3 Friendliness Analysis

In this section, the interaction of different TCP variants working with the proposed QoSatArt architecture is evaluated. This simulation test is set to see how a TCP variant can *live* besides each other variants in a shared network environment. Similar to the previous case, the satellite bottleneck link is considered to have constant capacity.

The simulation results showed on Fig. 5.2 correspond to the goodput evolution and the queue occupancy when considering the interaction of different TCP variants. In addition, Table 1 shows the number of acknowledged (acked) and retransmitted packets for each traffic class when considering the interaction of different TCP variants.

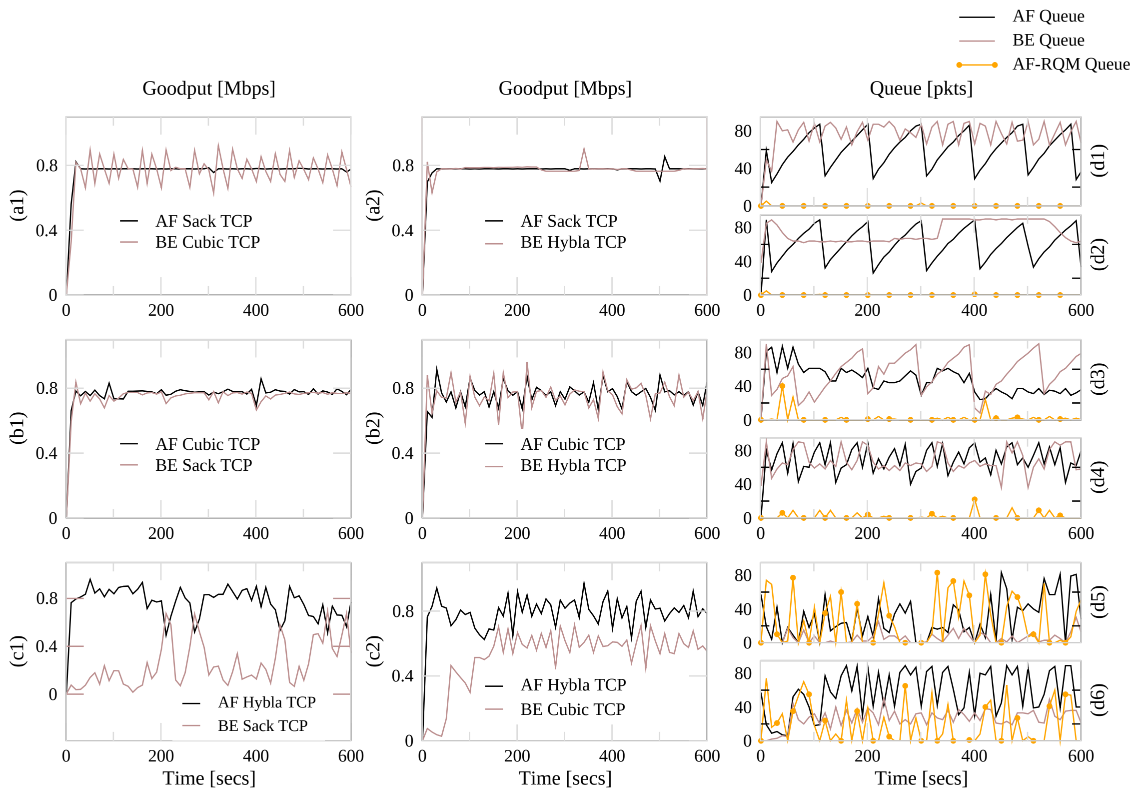


Figure 5.2: Goodput and queue occupancy results: Sack TCP interacting either with CUBIC TCP (a1) or Hybla TCP (a2), CUBIC TCP interacting either with Sack TCP(b1) or Hybla TCP (b2) and Hybla TCP interacting either with Sack (c1) or CUBIC TCP (c2).

It is worth mentioning that as the total received packets for the EF traffic class is kept as a constant value (51370 packets) considering all TCP interactions, the EF traffic class is not presented neither in the graphs nor in the Table 1 but it was included in the simulations.

Table 5.1: Number of acked and retransmitted packets for each traffic class considering the interaction of different TCP variants [packets]

TCP AF	TCP BE	AF acked [pkts]	BE acked [pkts]	AF retransmitted [pkts]	BE retransmitted [pkts]	Total [pkts]
Traffic class						
SACK	SACK	99,197	100,115	637	0	250,682
	CUBIC	99,582	100,063	365	1,411	251,015
	HYBLA	99,211	100,635	530	59,705	251,216
CUBIC	SACK	99,921	98,267	2,024	155	249,558
	CUBIC	99,915	99,125	1,336	1,246	250,410
	HYBLA	99,924	99,340	1,456	58,993	250,634
HYBLA	SACK	99,533	20,364	10,7164	839	171,267
	CUBIC	98,275	56,629	93,592	3,322	206,274
	HYBLA	99,135	68,005	162,947	43,815	218,510

On Fig. 5.2.a1 and Fig. 5.2.a2, the interaction of two flows are shown when Sack TCP is used to transport the AF traffic class and either CUBIC TCP or Hybla TCP is used to transport the BE traffic. As it is observed in both cases the TCP variants are able to fairly share the bandwidth, reaching their nominal 800 Kbps of goodput. In addition, in both cases (Fig. 5.2.a1 and 5.2.a2) the BE traffic is able to use the remaining bandwidth without any performance loss. The queue occupancy for this interaction is shown on Fig. 5.2.d1 and 5.2.d2. It is observed that both queues (the AF and BE queues) reach its BDP value which is set to 90 packets. In addition, the saw-tooth behavior is the result of using Sack TCP to transport the AF traffic class. In the same context, the friendliness level between Sack TCP and either CUBIC TCP or Hybla TCP is validated with the results showed in Table 1. As it is observed, each TCP variant is able to reach the similar number of acked packets (without retransmission) for each traffic class, reaching values between 99211 and 100635 acked packets.

In a similar way, Figs. 5.2.b1 and 5.2.b2 show the interaction when CUBIC TCP is used to transport the AF traffic class and either Sack TCP or Hybla TCP is used to transport BE traffic. In both cases, it is observed that each TCP variant is able to fairly share the bandwidth reaching its 800 Kbps of goodput. In addition, the AF traffic class preserves its level of priority during all the simulation. The queue occupancy for this interaction is shown on Figs. 5.2.d3 and 5.2.d4. As it is observed CUBIC TCP tries to keep the queue occupancy at low-medium levels when interacting with Sack TCP. Otherwise, when CUBIC TCP interacts with Hybla TCP, the queue occupancy of both flows reaches its upper limit (90 packets). This is mainly because two competing TCP variants that have a faster and more aggressive response (compared to Sack TCP) are used. The friendliness level when CUBIC TCP interacts either with Sack TCP or Hybla TCP is deduced from analyzing the results shown in Table 1. Here, both TCP flows are able to receive a similar amount of acked packets (between 99125 and 99924), mainly because CUBIC TCP was designed to interact with other TCP flows, thus allowing it to fairly share the available bandwidth between them.

Finally, Figs. 5.2.c1 and 6.c2 show the interaction when Hybla TCP is used to transport the AF traffic class and either CUBIC TCP or Sack TCP is used to transport the BE traffic. Here, the results in this interaction are extremely different compared to the previous cases. On Fig. 5.2.c1 when Hybla TCP interacts with Sack TCP, the former tries to reach its 800 Kbps of goodput, using more bandwidth than its nominal rate (800 Kbps). This situation results in an unfair bandwidth sharing, limiting the performance of the BE traffic class which is able to reach, in specific time points, peak values within the range of 600 Kbps. Similarly, when Hybla TCP interacts with CUBIC TCP (Fig. 5.2.c2), the former tries to reach again its 800 Kbps of goodput sending a great amount of packets. Although CUBIC TCP is able to reach an enhanced goodput level (600 Kbps in the long term) compared to Sack TCP, it is not able to reach its nominal value set to 800 Kbps. The oscillating behavior of this interaction occurs because both TCP variants have an aggressive response as a result of the congestion algorithm used in each case. The queue occupancy for this interaction is shown on Figs. 5.2.d5 and 6.d6. As it is observed, in both cases the BE queue is overloaded with a great number of retransmitted packets including the out-of-profile packets generated by the RQM mechanism. This situation directly affects the performance of the BE traffic class. The friendliness level when Hybla TCP interacts either with Sack TCP or CUBIC TCP is deduced from analyzing the results shown in Table 1. In all cases, the AF traffic class (using Hybla TCP) receives a higher number of acked packets (99533, 98275 and 99135) compared to the BE traffic class (20364, 56629 and 68005). In addition, the number of AF retransmitted packets for this interaction is highly increased (107164, 93592 and 162947) compared with the previous cases.

The results obtained with Hybla TCP are mainly caused by the combination of two factors: the aggressiveness of Hybla TCP (analyzed in Section 4.3.1) together with the implementation of the RQM mechanism. As the RQM mechanism allows to send out-of-profile packets from the AF queue to the BE queue, Hybla TCP increases the rate of retransmitted packets. Therefore, the queue occupancy is overloaded with a great number of retransmitted packets, affecting the goodput level for the BE traffic class.

In order to know the bandwidth allocation for each traffic class considering all the interactions Fig. 5.3 is presented.

As it can be seen, when Hybla TCP is used to transport the AF traffic class, more bandwidth resources are assigned compared to the cases when either CUBIC TCP or SACK TCP is used. These results lead us to avoid the use of Hybla TCP within the high priority classes, since this variant is unable to fairly share the bandwidth with lower priority classes.

Additionally, as it is observed on Fig. 5.3 when either Sack TCP or CUBIC TCP is used to transport the AF traffic class, the same amount of bandwidth resources are assigned, enhancing their friendliness level. Therefore, selecting these TCP variants would be desirable to transport high priority traffic. Particularly, for the case when CUBIC TCP is used to transport the AF traffic class, the simulation results enable us to observe an

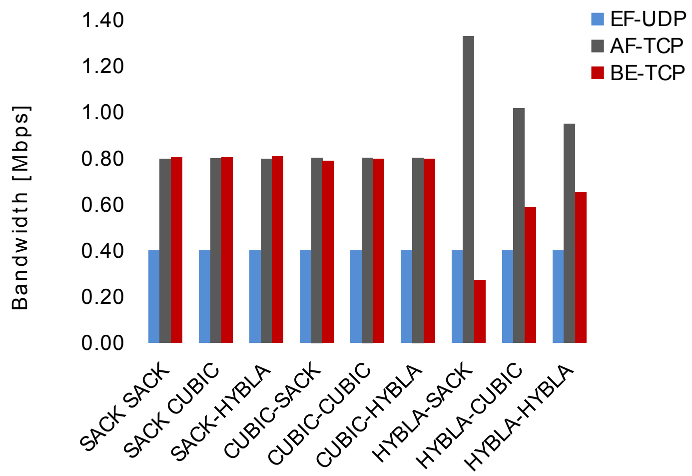


Figure 5.3: Bandwidth allocation for each traffic class considering the interaction of different TCP variants

enhanced friendliness level showing a less aggressive behavior compared to Hybla TCP. Moreover, as CUBIC TCP is able to keep the queue occupancy at medium levels (analyzed in graph 6.d3), the bandwidth utilization is optimized leading to a reduction of the system latency, being this a desirable situation when working with satellite systems. Considering the obtained results, in the next sections Hybla TCP has been discarded considering only Sack TCP and CUBIC TCP working in the context of the QoSatArt architecture.

5.4 Analysis of TCP variants considering traffic and bandwidth variations

The scenario analyzed in this section represents a common situation present over a DVB-S2 link, as it considers variations either in the traffic loads or the available bandwidth. To evaluate these characteristics affecting the TCP performance, two experiments are performed.

(i) The analysis of TCP variants considering that the EF traffic class suffers of load variations.

(ii) The analysis of CUBIC TCP considering bandwidth fluctuations. Here, the IP scheduler proposed for QoSatArt is compared against a RR and WRR based schedulers.

5.4.1 Analysis TCP variants considering EF traffic variations

This scenario analyzes the interaction between Sack TCP and CUBIC TCP working over a scenario that implements EF traffic variations. These EF variations represent Internet users demanding VoIP services at different time intervals. As a result, the bandwidth requirements of each traffic class will change as a function of time. In this experiment each traffic class starts transmitting at different time intervals: the BE traffic class starts at the beginning of the experiment, the AF traffic class starts after 100 seconds, while the EF traffic class starts transmitting after 200 seconds. To observe the effect of the EF traffic variation it is implemented in a periodic cycle.

Fig. 5.4 shows the simulation results when traffic fluctuations are present.

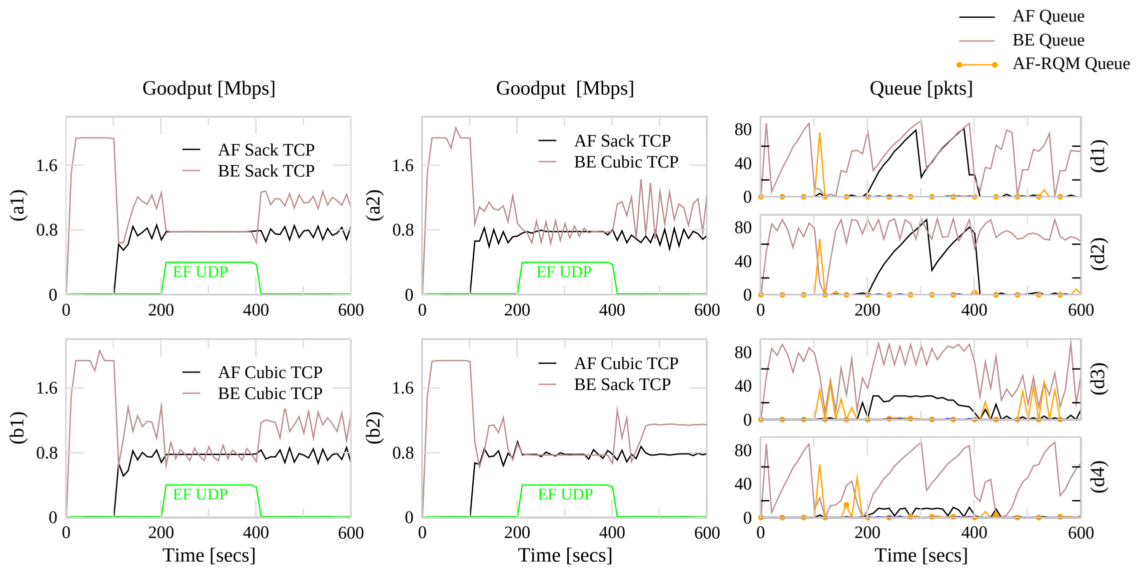


Figure 5.4: Goodput performance and queue occupancy considering EF traffic fluctuations between 200 and 400 sec in a periodic cycle

Particularly, Fig 5.4.a shows the goodput in Mbps when Sack TCP is used to transport the AF traffic class. On Fig. 5.4.b CUBIC TCP is used to transport AF traffic class. In all cases, both TCP variants are able to use the full satellite channel capacity (2 Mbps) at the beginning of the transmission. After 200 seconds, when the EF traffic starts transmitting through the network, the bandwidth for the BE traffic class is reduced to guarantee the EF traffic class. In all the cases, the AF traffic class keeps its level of goodput as a constant value of 800 Kbps.

In addition, when the EF traffic class is not present, the BE traffic class is able to use the remaining bandwidth that the EF traffic class has left. This expected result agrees with the QoSatArt objectives, since the EF traffic class has priority over the lower priority traffic classes. Moreover, in the same time interval where the EF traffic class is present (200-400 seconds), the ability of both TCP variants (Sack TCP and CUBIC TCP) to fairly share the rest of the available bandwidth is shown as both of them are able to reach its 800 Kbps of goodput.

Fig. 5.4.d (right graph) shows the queue occupancy levels when considering EF traffic variations. As it is observed, when using Sack TCP to transport the AF traffic class (Fig. 5.4.d1 and 5.4.d2), the queue occupancy reaches its limit set to 90 packets. Nevertheless, when CUBIC TCP is used to transport the AF traffic class, its ability to keep the queue occupancy at low levels is noticed even when the three traffic classes (EF, AF and BE) are presented. As it is seen on Fig. 5.4.d3 and 5.4.d4 within the interval between 200 and 400 seconds. This feature is essential to achieve optimum bandwidth utilization while keeping low system latency.

Summarizing, when the EF traffic class does not use a certain amount of bandwidth, the BE traffic is able to take advantage of this resource, therefore the proposed QoSatArt architecture successfully works in a scenario where EF traffic variations are present. In addition, it is observed that CUBIC TCP is the TCP variant that has an enhanced fairness level as it is able to fairly share the available bandwidth with other TCP variants such as Sack TCP. In addition, CUBIC TCP is able to achieve low latency since it keeps the queue occupancy at low-medium levels. Taking into account these conclusions, the next test considers only the CUBIC TCP variant (to transport the AF and BE traffic) over a bandwidth fluctuating scenario.

5.4.2 Analysis of CUBIC TCP considering bandwidth fluctuations.

The second experiment analyzes the interaction of two CUBIC TCP flows when the satellite link suffers degradations due to atmospheric conditions. Particularly, the evaluated QoSatArt satellite architecture (shown in Fig. 3.1) suffers of changes in the available link capacity due to the presence of a heavy rain event. Here, the rain event is simulated using a sinusoidal wave proposed in Section 4.2.2.

In particular, the QoSatArt architecture is evaluated and compared these results against two proposals. The first comparison includes the Round Robin mechanism as a packet scheduler together with a simple drop tail queue as a discarding mechanism. This is called a *RR-basic* configuration. Similarly, the second comparison considers the same drop tail queuing mechanism together with the Weighted Round Robin scheduler that uses static weight values set to μ_{EF}, μ_{AF} and μ_{BE} for the EF, AF and BE traffic classes respectively. This configuration is called a *WRR-static*.

Finally, a third comparison is performed considering our proposed QoSatArt design including the AQM system, the cross-layer IP scheduler and the RQM mechanism. As it is designed QoSatArt should enforce the priority levels taking into account the link capacity variations reported by the physical layer using the proposed cross-layer optimization. Therefore, the QoSatArt should be able to guarantee high priority traffic classes regardless of the weather and traffic load conditions.

Fig. 5.5 shows the results corresponding to the goodput evolution and queue occupancy when bandwidth variations are experienced.

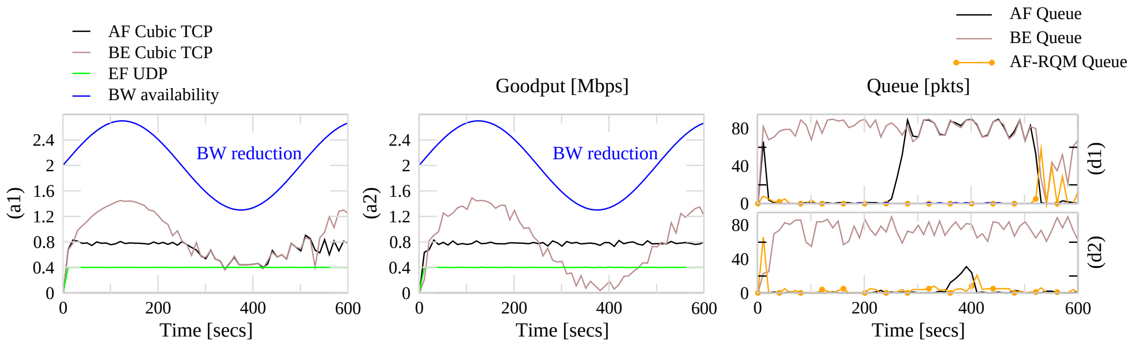


Figure 5.5: Goodput performance and queue occupancy results considering Cubic TCP in a bandwidth fluctuating scenario. (a1) Scenario considering RR scheduler, (a2) Scenario considering WRR scheduler and (d1-d2) Queue occupancy

Particularly Fig. 5.5.a1 shows the CUBIC TCP goodput level when the *RR-basic* architecture is used. From these results, it is observed that the RR mechanism is not able to guarantee the transmission rate for the AF traffic class when a bandwidth reduction is present (see the valley of the sinusoidal wave). Similarly, the AF queue occupancy, shown on Fig 5.5.d1, reaches its maximum level (90 packets) when a bandwidth reduction is experienced (see the interval between 200 and 600 secs).

The main problem occurs when the link bandwidth is considerably reduced and as a consequence the queue occupancy starts overloading. This situation is mainly because the RR mechanism extracts a packet every time it visits each queue. As a result, all the queues receive the same treatment. This defined priority policy does not work properly given that all the classes obtain the same bandwidth. To address this problem, the implementation of a WRR algorithm in which queues of the higher priority classes become much more visited than the BE queue. In particular, the WRR weights were selected as 7, 7, and 1 for the EF, AF and BE traffic classes respectively according to the guidelines defined in [RMMDA+12b].

Fig. 5.5.a2 and 5.5.d2 show the goodput and the queue occupancy results respectively, when the *WRR-static* configuration is considered. From these results it is possible to observe that the EF and AF traffic classes are always guaranteed, even though a severe bandwidth reduction is present. Particularly, when the bandwidth availability is reduced (see the valley of the sine wave), the AF traffic class is guaranteed, being able to reach its nominal rate of 800 Kbps, while the BE traffic class is penalized, reaching only 100 Kbps. In contrast to the case when the link capacity is increased (see the peak of the sine wave), the AF traffic class keeps its nominal rate of 800 Kbps, while the BE traffic class uses the remaining bandwidth.

In this context, the AF queue occupancy using CUBIC TCP, is always kept at lower levels, as it is seen on Fig. 5.5.d2, despite the fact that the satellite systems is working on a reduced capacity.

5.4.3 QoSatArt: CUBIC, Sack and Compound TCP

Finally, to have a general conclusion about the CUBIC TCP performance an additional test has been proposed. The objective is to compare the performance of CUBIC TCP against other TCP variants working in the QoSatArt scenario in which a heavy rain event is affecting the DVB-S2 satellite link. In this experiment Sack TCP is added again, as it represents a baseline TCP variant. In addition, Compound TCP [TSZS06] has been included, as it is characterized for having an enhanced congestion control algorithm and also for being the native transport protocol of Windows 7. This TCP variant is a modified version of Reno TCP in which a scalable delay-based component is added into the standard TCP Reno congestion avoidance algorithm. Compound TCP has the ability to perform estimations of queuing delay as a measure of congestion. In this way, when the queuing delay is small, this TCP variant assumes that the links are not congested, and therefore it will rapidly increase its window growth rate.

The simulation results of goodput and queue occupancy considering Sack, Compound and CUBIC TCP when a heavy rain event is affecting the DVB-S2 satellite link are shown on Fig. 5.6. Particularly, Fig. 5.6.a1 and b1 and c1 show the goodput results, using Sack TCP, Compound TCP and CUBIC TCP respectively. In these figures, it is possible to observe that using different TCP variants, the AF traffic class is always guaranteed over the BE traffic class, even though a severe bandwidth reduction is present in the satellite system (see interval between 300 to 400 seconds).

This is mainly because the proposed QoSatArt architecture allows to guarantee high priority traffic classes. Moreover, in the cases when using either Sack TCP or Compound TCP (5.6.a1, b1) both AF goodput patterns behaves similarly. This response is obtained because both TCP variants design are based on the Reno TCP and therefore the classical *cwnd*'s Reno saw-tooth behavior is observed, resulting thus in a oscillating goodput. Nevertheless, when Cubic TCP is used (5.6.c1), a more stable goodput pattern is shown, as a result of using a congestion window algorithm based on a cubic function.

In the same context, the AF queue occupancy results, using Sack TCP, Compound TCP and CUBIC TCP, are shown on Fig. 5.6.a2, b2 and c2.

In these graphs it is possible to observe that Sack TCP has the highest level of queue occupancy with values ranging between 58 packets. Similarly, on Fig. 5.6.b2 Compound TCP reaches queue occupancy values ranging between 53 packets. Although this last value is smaller compared to Sack TCP, it is higher when compared to CUBIC TCP which is able to buffer 25 packets in its queue.

As a result, CUBIC TCP is able to provide a stable goodput level for the AF traffic class while keeping the queue occupancy at lower levels when a DVB-S2 satellite system experiences a heavy rain event.

Summarizing the simulation results presented in this subsection, it can be concluded that when the proposed QoSatArt works together with the adoption of CUBIC TCP variant, it is possible to obtain an enhanced TCP performance when the satellite link experiences severe bandwidth reductions due to the presence of rain events.

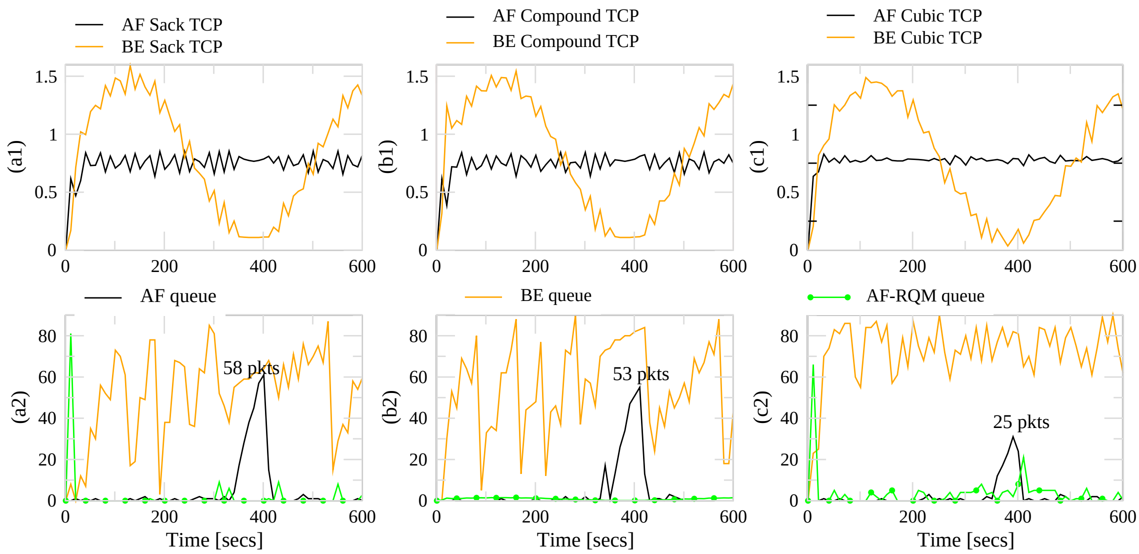


Figure 5.6: Goodput performance and queue occupancy results considering Sack TCP (a1,a2), Compound TCP (b1,b2) and Cubic TCP (c1,c2) in a bandwidth fluctuating scenario.

5.5 Conclusions

In this chapter an analysis of several TCP variants has been performed in the context of a QoSatArt scenario. The QoSatArt proposes an architecture developed on the SI layer, particularly, on the network layer to provide QoS guarantees.

The TCP analysis has been performed using metrics such as goodput, queue occupancy, delay and jitter values in order to analyze the friendliness and fairness behavior among the interaction of several TCP variants.

Simulation results have shown that Hybla TCP shows an aggressive behavior which adversely affects the overall satellite performance. This is mainly caused because of the implementation of the RQM mechanism (proposed in QoSatArt Section 3.4.2) that enables the allocation of a considerable amount of out-of-profile packets in the BE queue.

On the other hand, Sack TCP shows good fairness and friendliness level, as it is able to fairly share the bandwidth when constant link capacity is considered.

CUBIC TCP is able to provide a stable goodput level for the AF traffic class while keeping the queue occupancy at lower levels when a DVB-S2 satellite system experiences bandwidth reductions.

Finally, with the simulation results presented in this chapter, it can be concluded that when the proposed QoSatArt works together with the adoption of CUBIC TCP variant, it is possible to obtain an enhanced TCP performance when the satellite link experiences severe bandwidth reductions due to the presence of rain events while guaranteeing the required QoS specification on the whole E2E path.

Taking into account the obtained results from the evaluation of standard TCP versions, the next step in the research work considers the possibility to design a modified TCP variant that manage efficiently the TCP connections while dealing with QoS issues and the intrinsic characteristics present in GEO satellite systems. Therefore in the next chapter a proposal that considers a new TCP version based on a cross-layer design has been develop.

CHAPTER 6

XPLIT-FW a cross-layer PEP architecture for the forward channel

TCP modifications have been investigated in order to analyze the problems introduced by large Bandwidth-Delay product present in satellite systems, in which parameters and algorithms tunning have been developed. Nonetheless, the benefits that these modifications are able to bring are limited since they do not rely on the detailed acknowledgement of the satellite environment and of the protocols acting below TCP.

In this context, the implementations of intermediate gateways able to manage efficiently TCP connections with the adoption of mechanism aiming to translate and modify the transport protocol format are necessary. Here, the Performance Enhanced Proxies (PEPs) represent a suitable option for the design of QoS gateways that implements mechanisms at any level of the protocol stack from the transport layer down to the physical layer. Using PEPs the TCP connection is divided into a satellite portion and a terrestrial portion in order to isolate the long latency satellite link. PEPs are responsible for intercepting, catching and acknowledging data received by senders and forwarding these data to the receivers.

As it was previously mentioned, the long propagation delay, present in GEO satellite systems, severely affects the TCP performance due to the intrinsic TCP design which is determined based on the congestion window (*cwnd*) size.

In order to overcome TCP limitations, some authors have proposed approaches with some degree of explicit knowledge about the network state, allowing a better optimization of the TCP performance. This type of approaches are called *cross-layer* TCPs [KGC07], in which the lower layers send feedback information to the TCP layer in order to enhance its performance. This is done by parameterizing the TCP congestion control mechanism including the slow start, congestion avoidance, fast retransmit and fast recovery phases and the parameters associated to each phase.

In this chapter, the XPLIT architecture developed for the DVB-S2 forward channel is developed. It is based on PEPs and cross-layer design, following the guidelines defined in the ETSI-BSM-QoS standard. The design includes the development of a cross-layer

TCP called XPLIT-TCP, a new satellite-optimized TCP protocol that uses cross-layer information reported by lower layers to improve the performance of TCP flows in the satellite segment.

6.1 XPLIT-FWD scenario

The XPLIT architecture is based on splitting and cross-layer design, this has been developed for the forward channel considering a DVB-S2/ETSI-BSM QoS scenario. The scenario is shown in Fig. 6.1 which includes a Broadband GEO satellite system working in the Ka frequency band.

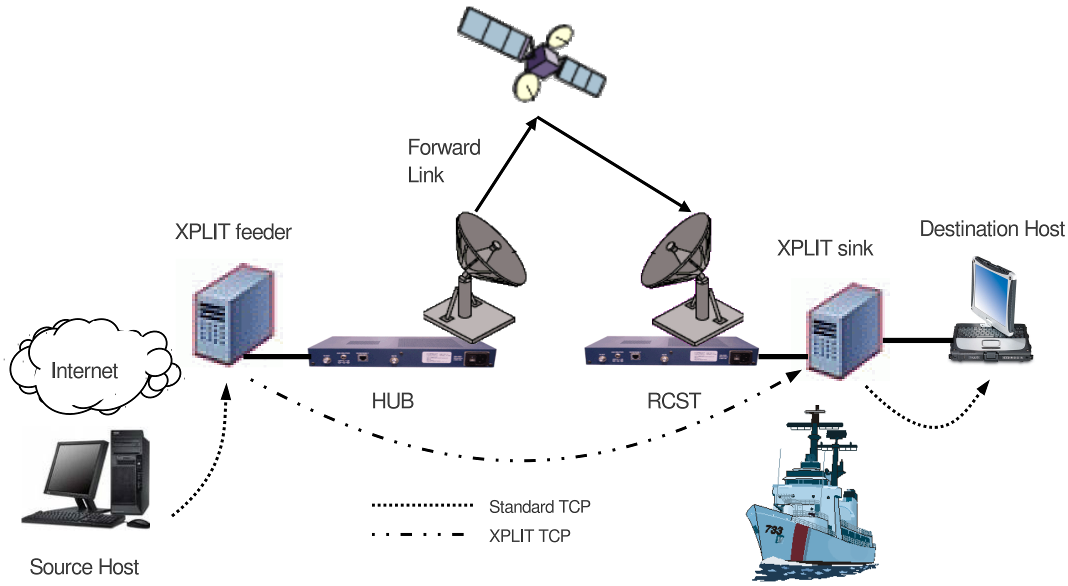


Figure 6.1: Scenario for the DVB-S2 XPLIT architecture

As it can be observed, there are two elements that manage the physical medium and the data link layer of the satellite link: a satellite forwarder node (HUB) and a Return Channel Satellite Terminal (RCST). Each satellite subscriber has a RCST, while the satellite operator manages the HUB. The link HUB-to-RCST is called *forward* link and it is defined by the DVB-S2 standard [ets04]. The RCST-to-HUB link is called *return channel* and it is defined by the DVB-RCS standard [ets05a]. In practice, the HUB is located in the infrastructure side (which belongs to the operator) where it is also connected to the Internet. Therefore, the forward link is intensively used by satellite terminals in their access to Internet applications and services.

PEP boxes are also included to deploy the TCP-splitting architecture, connecting the HUB to the Internet access and the RCST terminals to the Destination Host. Here, the direction of the data flow considers a PEP feeder attached to the HUB at the infrastructure side, this PEP will feed the satellite link with data. Similarly, the PEP boxes attached to RCST terminals are called PEP sinks which are responsible for receiving data coming from the GEO satellite.

6.2 XPLIT architecture design

The XPLIT-TCP protocol is a new satellite-optimized TCP protocol that uses cross-layer information to improve the performance of TCP flows in the satellite segment. The cross-layer information used by XPLIT-TCP is the buffer occupancy and the service rate (β_i and μ_i). It is worth mentioning that in [AMDMn$^+$12] and [RMMDA$^+$09], the complete design of the XPLIT architecture is developed,together with an extensive performance evaluation of the XPLIT-TCP variant.

The XPLIT-TCP design is based on the traditional TCP design, as it implements the standard TCP headers, the TCP standard flow control (based on a sliding window) and the TCP standard error control (based on retransmissions and time-outs). Nevertheless, the XPLIT-TCP does not include probing phases as congestion control mechanisms such as slow start or congestion avoidance. In this case, it uses two control loops to manage the congestion changes in the satellite system. The first control loop uses periodic polling of the cross-layer information β_i and μ_i to regulate the slow dynamics of the system (i.e. new TCP connections) and a second control loop to manage the fast and sudden dynamics which are mainly due to changes of available service rate for a traffic class (i.e. burst of packets).

In this way, the XPLIT-TCP uses the information reported by cross-layer to adjust the congestion window value of the active senders according to the efficiency and fairness conditions measured at each moment.

The XPLIT architecture is depicted on Fig. 6.2. As it is shown, incoming flows are received by the PEP feeder to be forwarded through the HUB to the destination RCST.

The architecture includes *XPLIT feeders* allocated at the PEP feeder that send data to the *PEP sink* allocated at the destination PEP, using the TCP protocol. The HUB and the XPLIT feeder are connected through a Local Area Network (LAN).

Inside the PEP feeder, the *XPLIT manager* is responsible for creating a buffer for each incoming flow represented by the combination *TCP receiver*- *XPLIT feeder*. It sends the data to the *PEP sink*

The TCP segments of each incoming flow are delivered to the corresponding *TCP receiver* allocated either at the PEP sink or directly at any host attached to the RCST (see Fig. 6.1).

Using this PEP-based architecture, it is possible to run the satellite-optimized TCP protocol XPLIT-TCP at both source and destination Hosts.

Particularly, XPLIT-TCP is centrally managed by the *XPLIT manager*, which initiates the *XPLIT senders* and manages them during their lifetimes.

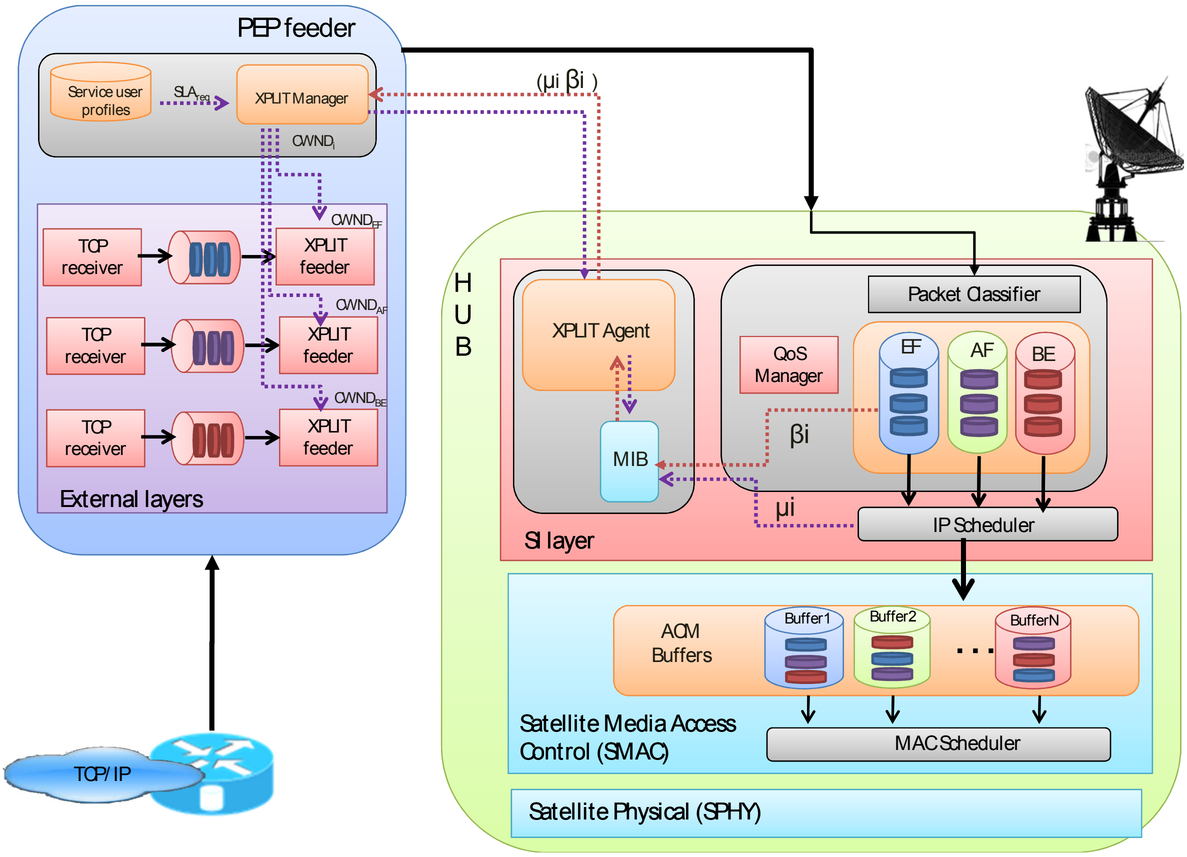


Figure 6.2: XPLIT architecture.

XPLIT-TCP uses the standard TCP headers together with a modified congestion control algorithm. Using standard TCP headers it is possible to able *XPLIT senders* compatible with standard *TCP receivers*

In particular, the *XPLIT manager* (allocated at the PEP) computes the flight-size packets available for each flow using the cross-layer feedback information (β_i and μ_i) reported at the XPLIT signaler. It externally monitors and manage the occupancy of the different Diffserv classes allowing to fair sharing of the capacity allocated among aggregated flows. It also considers the fairness policy to distribute this flight-size among the active flows of each traffic class.

To provide QoS guarantees the XPLIT architecture is designed based on the DiffServ framework. In which packet classification, marking and the definition of the DSCP field are performed by the edge router before reaching the *XPLIT feeder* (see Fig. 6.1). Here, for each PHB i, the service rate μ_i and the buffer occupancy β_i parameters are monitored by the *XPLIT manager*.

In particular, the service rate μ_i is computed by the IP scheduler (see also Fig. 6.2), which controls the number of packets that have been extracted out from its queues. This parameter, together with the buffer occupancy level of the each Diffserv queues are stored at the *Xlayer MIB* (Management Information Base).

Here, the *XPLIT signaler* play a crucial role to guarantee the interchange of cross-layer information as it sends this feedback to the *XPLIT manager*.

Therefore the cross-layer information flows from the MAC layer at the HUB to the Transport layer at the PEP.

It is worth mentioning that the signaling information is locally exchanged without crossing the satellite link interface, achieving a negligible signaling overhead and a low signaling delay. In addition, XPLIT architecture does not require to be aware about the assignment of physical/MAC resources to traffic classes at the satellite link since this task is performed by the ETSI-BSM QoS architecture.

To reach full satellite network exploitation XPLIT architecture is able to know and update the currently system load (i.e. the number of packets sent but not yet acknowledged) by using the cross-layer information: the buffer occupancy and the service rate (β_i and μ_i). Therefore, congestion losses are avoided by design while transmission losses are managed to do not consider packet injection rate reductions preventing congestion epochs. Based in this, XPLIT-TCP algorithm continuously manages the system load to avoid congestion looses.

To provide a minimum user response time XPLIT-TCP does not implement probing phases. This is done given that the currently available bandwidth is known and updated by the cross-layer design. Therefore, the XPLIT architecture is able to manage the buffer occupancy level to enforce the SLA specifications of each Diffserv class.

To provide an efficient resource sharing, the XPLIT-TCP decouples functions related to efficiency (maximizing link utilization) from functions related to fairness (available resource distribution).

Here, the *XPLIT manager* manages and monitors each aggregated Diffserv traffic that enters in a queue. The scalability of the algorithm is ensured as it is not necessary to process per flow cross-layer information, allowing to keep the signaling overhead at lower levels. The efficiency function of XPLIT-TCP manages the total system load of each traffic class in order to keep the suitable working point that maximizes the link utilization, while fulfilling the QoS level specification.

On the other hand, to provide fair allocation of resources. XPLIT architecture applies the selected policy to share the available bandwidth (aggregated flight-size) among competing flows. The fairness distribution is modified every time either the system load (which is time-dependent) or the number of active flows change.

6.2.1 Signaling mechanism design

Fig. 6.3 illustrates the signaling mechanism used in XPLIT architecture. Similar to QoSatArt architecture, it is based on the SNMP protocol. Here, the QoS parameters are determined by the SLA, and any SLA variation should be handled by a SLA negotiation.

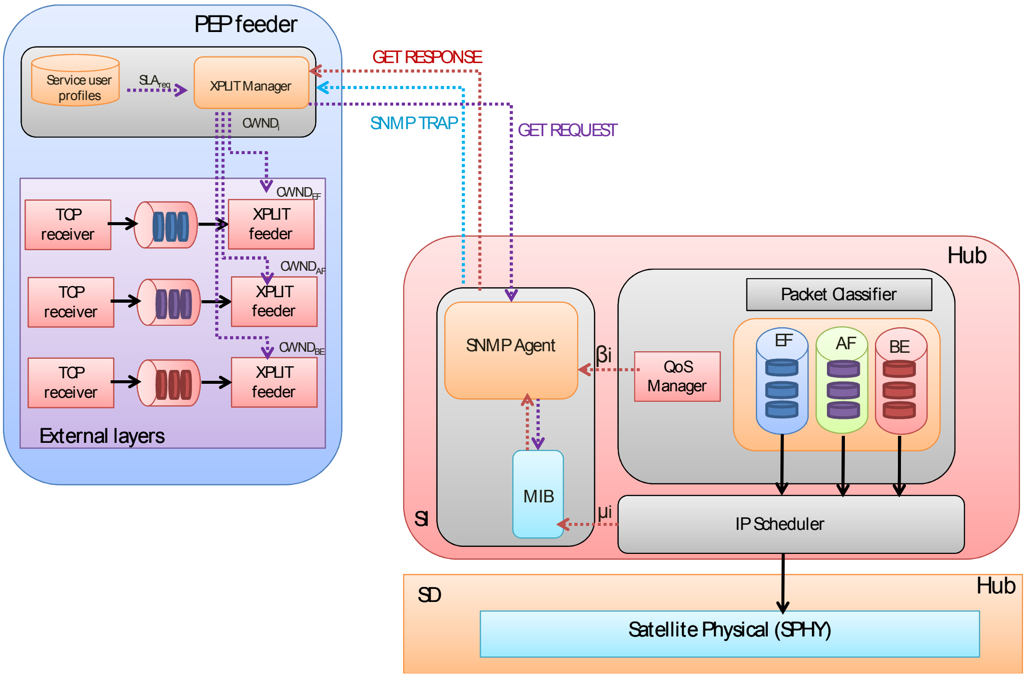


Figure 6.3: XPLIT Signaling Mechanism.

In XPLIT architecture, it is assumed that the user terminal is capable of signaling and managing IP QoS resource request. However, it requires prior authorization from the Service Controller Server. The RCST terminal is responsible for requesting an application-specific service by sending a *Service request* to the Controller. The service controller is in charge of determining the QoS needs of the requested service.

In XPLIT architecture, the QoS signaling messages are sent using out-of-band signaling. Here, the the Single Network Management Protocol (SNMP) [HPW02] is proposed to transport QoS signaling. This protocol allows defining by means of the Structure Management Information (SMI) [Pre02] the allocation the cross-layer information using the Management Information Base (MIB) unit.

The SNMP protocol is a standard management protocol which is commonly used on networking devices with open source implementations. The SNMP can work over the TCP protocol providing a reliable and secure signaling service while taking advantage of the event-driven mechanism to send QoS signaling.

As it is shown on Fig. 6.3 the *SNMP manager* is responsible for monitoring and managing network devices at the Management plane (M-plane). Each *SNMP manager* interacts with an *agent* to report signaling information.

The *SNMP manager* is allocated at the PEP feeder while the *agent* processes are allocated at the DVB-S2 gateway.

SNMP offers two ways of exchanging information: traps and polling. When using polling, To request information the *SNMP manager* uses `Get-Request` messages. The *agent* should respond using `Get-Response` messages.

Alternatively, when an event occurs on the device in which the manager must be aware, the *agent* should send a `Trap` message to the manager, indicating the variables that have changed.

In XPLIT, the *XL-Manager* periodically polls the bandwidth availability C_{OUT} stored in the MIB The *signaler agent* responds to the *XL-Manager* sending a `Get-Response` message with the updated values.

6.3 XPLIT-TCP Congestion Control mechanism

The XPLIT-TCP protocol sets the congestion window value externally provided by the cross-layer XPLIT architecture. XPLIT-TCP uses a congestion control algorithm that implements two control loops in order to manage variations of the system parameters.

XPLIT-TCP uses a window-based algorithm to ensure network stability while reducing bandwidth estimations sensitivity.

6.3.1 System model and analysis

The design the XPLIT congestion control algorithm considers the queue model depicted in Fig. 6.4. It considers a single Diffserv queue of the Satellite Independent (SI) network layer. The SI queues are controlled by the QoS IP scheduler which continuously decides the assigned service rate to each class, allowing to manage Diffserv classes separately.

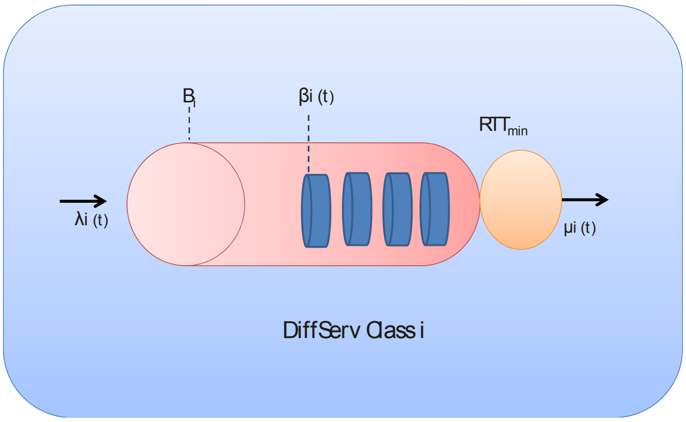


Figure 6.4: System model for a SI queue

Here, $\lambda_i(t)$ is the packet arrival rate for traffic class i, $\mu_i(t)$ is the service rate for traffic class i, $\beta_i(t)$ is the level of occupancy of queue i, B_i is the size of queue i and RTT_{min} is the two-way propagation delay or minimum Round Trip Time[1].

[1]The RTT_{min} is around 520 ms in the GEO satellite scenario.

The system load is expressed by the number of packets that have been sent but not yet acknowledged, for each Diffserv traffic class i at instant t (see Eq. 6.1):

$$L_i(t) = \mu_i(t) \cdot RTT_{min} + \beta_i(t) = BDP_i(t) + \beta_i(t) \tag{6.1}$$

Where the Bandwidth-Delay-Product (BDP) is set by the product of the service rate and the RTT_{min}. On the other hand, the maximum system load available for a traffic class i at instant t is:

$$L_i^{max}(t) = \mu_i(t) \cdot RTT_{min} + B_i \tag{6.2}$$

Basically, the XPLIT-TCP congestion control algorithm regulates the congestion window of the senders of each traffic class to enforce the system to work at a reference load level defined by $L_i^{ref}(t)$. This reference level or working point is expressed as:

$$L_i^{ref}(t) = \mu_i(t) \cdot RTT_{min} + \beta_i^{ref} \quad 0 \leq \beta_i^{ref} \leq B_i \tag{6.3}$$

Where $L_i^{ref}(t) \leq L_i^{max}(t)$, and β_i^{ref} are calculated as a function of the SLA requirements of the Diffserv class i. Eq. 6.3 also represents the desired in flight-size value. In this way, the number of packets of the traffic class i is able to keep the buffer occupancy at the reference level β_i^{ref} in order to have a complete link utilization. This efficiency function of XPLIT-TCP manages the total system load of each traffic class in order to keep the established working point $L_i^{ref}(t)$. .

On the other hand, the XPLIT architecture is able to apply any desired SLA policy to share the aggregated flight-size available. One strategy to share the system load is to equally distribute the available bandwidth among the active flows. This is done every time either the system load or the number of active flows change.

Therefore, if there are $N_i(t)$ active connections of traffic class i at instant t, then the congestion window of each active flow $cwnd_i(t)$ fulfill the following equation:

$$L_i^{ref}(t) = N_i(t) \cdot cwnd_i(t) \tag{6.4}$$

And therefore,

$$cwnd_i(t) = \frac{L_i^{ref}(t)}{N_i(t)} = \frac{\mu_i(t) \cdot RTT_{min} + \beta_i^{ref}}{N_i(t)} \tag{6.5}$$

As a result the $cwnd_i(t)$ value represents the congestion window per flow, which is the the control signal to manage the system load. In this particular case this value is the same for all the flows as the system load is equally distributed.

However, when using $cwnd_i(t)$ as control signal it is important to consider that sudden increments of this window will generate burst of packets. Although the design of the congestion control algorithm is able to tailor the allocated resources in the steady state, in the transient state the HUB could receive a burst of packets generating buffer overflows.

To solve this in XPLIT architecture, the use of packet pacing as rate control mechanism is proposed.

6.3.2 Rate control mechanism

In this case, a delay is added between the transmission of successive packets. This delay, noted as $\delta_i(t)$, is called *spreading time* which is time-dependent and class-dependent. This value is set considering the worst case situation in which the system queue is empty. In this situation, the system must be fill with packets in order to reach the desired working point $L_i^{ref}(t)$ considering the RTT_{min}.

Therefore, the maximum rate required of an aggregated packet arrival is set by Eq. 6.6:

$$\lambda_i^{max}(t) = \frac{L_i^{ref}(t)}{RTT_{min}} = \frac{N_i(t) \cdot cwnd_i(t)}{RTT_{min}} \tag{6.6}$$

In this situation, the satellite system is able to work in self-clocking stage during a RTT_{min}. Using $\lambda_i^{max}(t)$ it is possible to fill the link capacity assigned to the class i completely, while keeping the corresponding DiffServ queue close to the β_i^{ref} reference level during a RTT_{min}.

Finally, the inverse of the spreading time per flow is set equal to the maximum packet arrival rate per flow, which is the maximum aggregated packet arrival rate divided by the number of active flows:

$$\frac{1}{\delta_i(t)} = \frac{\lambda_i^{max}(t)}{N_i(t)} \tag{6.7}$$

From (6.5), (6.6) and (6.7), it is possible to obtain the expression for the *spreading time* for the flows of class i:

$$\delta_i(t) = \frac{RTT_{min}}{cwnd_i(t)} = \frac{RTT_{min}}{(\mu_i(t) \cdot RTT_{min} + \beta_i^{ref})/N_i(t)} \tag{6.8}$$

Here, it is important to consider that small variations in the two-way propagation delay RTT_{min} generates similar variations of the system load. As a result the working point $L_i^{ref}(t)$ in the satellite system defined by Eq. (6.5) must consider these changes.

To complement the XPLIT-TCP design, two different control loops (not performed simultaneously) in the system are considered:

(i) The *Periodic Management* control loop is defined to monitor the current system load. It compares this value with the desired reference load and performs the necessary actions to modify if necessary the system parameters.

(ii) The *Event Management* control loop is defined to perform system modifications triggered by specific events, such as important service rate variations or changes in the number of active flows.

6.3.3 Periodic Management

In a real DVB-S2/ETSI-BSM QoS system there are some phenomena that cannot be parameterized and thus they cannot be directly analyzed by the proposed system model. These situations generates errors in defining the buffer occupancy level of the Diffserv queues.

For instance, smooth changes in the satellite orbit generates slow variations of the two-way propagation delay, which in turn introduce small disturbances of the working point $L_i^{ref}(t)$.

On the other hand, either inaccurate measurements or slow variations of the service rate due to gradual changes of ModCods can also cause small disturbances of the working point $L_i^{ref}(t)$.

Furthermore, unregulated traffic generated by Customer Premises Equipment (CPEs) in the forward link (i.e ACK packets) requires a small fraction of the available resources.

The main effect that these situation generates in the proposed model is that the buffer occupancy is not able to be positioned at its reference value β_i^{ref}.

In order to correct these queue position errors $\epsilon_i(t)$, the proposed design periodically polls the buffer occupancy $\beta_i(t)$ to perform correction actions if necessary.

Let us assume that the polling interval T_{poll} occurs at instant n. Then, the queue position error can be defined in terms of n as:

$$\epsilon_i(n) = \beta_i(n) - \beta_i^{ref} \tag{6.9}$$

Therefore, if there is a queue position error, the efficiency controller corrects the previous reference load level to compensate the observed error. Then, the fairness controller distributes the updated system load among the active flows of the class. Summarizing, the correction action carried out over flows at the $n-th$ instant of polling can be calculated as follows:

$$cwnd_i(n) = cwnd_i(n-1) - \frac{\epsilon_i(n)}{N_i(n)} \tag{6.10}$$

The previous window correction restores the buffer occupancy to β_i^{ref}. Notice that if $\beta_i(n) = \beta_i^{ref}$, then the window correction is not necessary because $cwnd_i(n) = cwnd_i(n-1)$. But if $\beta_i(n) \neq \beta_i^{ref}$, this means that the system is experiencing some unparameterized phenomena necessary to correct.

An important issue now is which is the suitable T_{poll} for the algorithm. T_{poll} must be chosen in such a way that two successive control actions do not interfere with each other. Next, we are going to show that it is enough to take a T_{poll} greater or equal than RTT_{min}. This will assure that the system has reached the working point according to the last control action performed. To show this, we have to distinguish between two cases because the system exhibits different dynamics for increments and decrements of the congestion window.

The first case is when $\epsilon_i(n) < 0$, in which the queue is emptying below β_i^{ref}. According to Equation (6.10), the control action determines that the congestion window has to be increased to make the queue reach again β_i^{ref}, so $cwnd_i(n) > cwnd_i(n-1)$. This means that the senders must increase the system load. As previously mentioned, our algorithm is designed with packet pacing to avoid packet bursts. Anyway, our packet pacing mechanism is designed for the worst case possible in which we have to introduce all the system load in a RTT_{min}. Therefore, any fraction of the total load can be restored in less time. Hence, we can consider RTT_{min} as the lower bound for T_{poll}.

The second case is when $\epsilon_i(n) > 0$, in which the queue is filling beyond β_i^{ref}. According to Equation (6.10) the control action determines that the congestion window has to be decreased, so $cwnd_i(n) < cwnd_i(n-1)$. This means that we must decrease the system load.

However, increasing and decreasing the congestion window are not symmetric operations. While in the former we use the spreading factor to increase the packet arrival rate $\lambda_i(t)$, in the latter the packet arrival rate is enforced to be zero until the packets that caused the queue position error $\epsilon_i(n)$ leave the corresponding queue. Notice that these packets will be able to leave the queue and get into the link in an RTT_{min} if:

$$\epsilon_i(n) < BDP_i(n) \tag{6.11}$$

On the other hand, the maximum positive error between two polling instants is bounded by the physical limits of the corresponding queue:

$$\epsilon_i^{max} = B_i - \beta_i^{ref} \tag{6.12}$$

Therefore, to meet Equation (6.11) in any case, the maximum possible error has to be lower than the minimum possible BDP:

$$\epsilon_i^{max} = B_i - \beta_i^{ref} < BDP_i^{\ min} \tag{6.13}$$

From the previous expression, we can define a minimum service rate μ_i^{min}:

$$\mu_i^{min} = \frac{B_i - \beta_i^{ref}}{RTT_{min}} \tag{6.14}$$

If $\mu_i(t) > \mu_i^{min}$, we can assure that exceeding packets will leave their corresponding queue in less than an RTT_{min}. Thus, in this case it is also enough to set T_{poll} to one RTT_{min}.

Finally, there is a special case that is when $\mu_i(t) < \mu_i^{min}$. In this case, we can consider that the queue is stalled, and we need to define an specific procedure to handle this situation. This procedure is as follows:

1. First, when the system detects that $\mu_i(t) < \mu_i^{min}$, the reference system load is recalculated following Equation (6.3), which yields the following condition:

$$L_i^{ref}(t)\big|_{\mu_i(t)\leq\mu_i^{min}} \leq \mu_i^{min} \cdot RTT_{min} + \beta_i^{ref} \tag{6.15}$$

2. Next, the new system load is distributed among active flows following Equation (6.5).

3. Then, the polling control loop is deactivated because while the queue is stalled it cannot be assured that the system has reached the working point according to the last control action performed.

4. Finally, there are not corrective actions until the queue is served above the minimum service rate.

An important additional result that we can assure by construction of our algorithm is that there is not going to be congestion (buffer overflow) while the queue is stalled. This is demonstrated applying Equation (6.14) to Condition (6.15), which yields:

$$L_i^{ref}(t)\big|_{\mu_i(t)\leq\mu_i^{min}} \leq B_i \tag{6.16}$$

As it is shown in the previous equation, when the queue is stalled, the system load of each Diffserv class can be always accommodated within its corresponding queue. So then, summarizing the previous discussion, we set T_{poll} to be equal to RTT_{min} and we activate or deactivate the polling mechanism after the service rate goes above or below the threshold μ_i^{min}.

To complete the design of the periodic management, there is still an important issue that has to be considered. We can correct slow variations caused by the unparameterized phenomena with polling. However, a relevant change in the system load between polling intervals cannot be properly managed if the system load variation is big enough. In this case, if the XPLIT manager is not aware of this situation, we might have buffer overflows (congestion) or buffer underflows (not full link exploitation).

The parameter that can cause a critical and sudden change of the system load is the service rate in a DVB-S2/ETSI-BSM QoS scenario. Let us consider that we are at the beginning of the $n-th$ polling interval, and at this precise moment, the system is regulated to β_i^{ref} and $\mu_i(n)$. Let us consider the worse case, in which the service rate changes to $\mu_i \to \mu_i + \Delta\mu_i$ just after applying the window correction. Then, we will have two critical bounds for $\Delta\mu_i$:

- $\Delta\mu_i^u$, which is an increment of $\Delta\mu_i$ that will cause a buffer underflow because the actual service rate is faster.

- $-\Delta\mu_i^o$, which is a decrement of $\Delta\mu_i$ that will cause a buffer overflow because the actual service rate is slower.

To calculate these bounds, we have to remember that the maximum excess of packets that we can allow to be accumulated during a polling interval is $B_i - \beta_i^{ref}$, while the minimum deficit of packets that we can allow during a polling interval is $-\beta_i^{ref}$. Expressing these conditions over the position error:

$$-\beta_i^{ref} \leq \epsilon_i(n) \leq B_i - \beta_i^{ref} \quad \forall n \tag{6.17}$$

Considering $T_{poll} = RTT_{min}$, we can obtain the critical bounds for the service rate variations:

$$\Delta\mu_i^u = \frac{\beta_i^{ref}}{RTT_{min}} \tag{6.18}$$

$$\Delta\mu_i^o = \frac{B_i - \beta_i^{ref}}{RTT_{min}} \tag{6.19}$$

The previous bounds define the maximum variations of the service rate allowed while using polling[2]. Service rates variations above the critical bounds must trigger a trap message to start the event management control loop for quick reaction, which is described in the next section.

6.3.4 Event Management

We define an *event* as a significant change of the system settings. In particular, XPLIT considers two events: a variation of the service rate that triggers a trap message, and a modification in the number of active connections. The system should be informed about these events to avoid possible buffer overflows or underflows, and to fairly exploit available resources. The reaction to an event is a resizing of the windows of senders according to the new parameters (whose nominal values are explicitly known).

In the first case, a service rate variation that triggers a trap message, the correction action consists in computing the new reference load level $L_i^{ref}(t)$ (according to Equation (6.3)) and then, resizing the window (according to Equation (6.5)). After this resizing, the new window can be increased or decreased. The question now is that we need to know which is the minimum time that we have to wait before we can restart the periodic management. This is necessary because we need to ensure that the system is properly set before applying any new control action.

As previously mentioned, window increases and window decreases are not symmetric operations in window-based rate congestion control algorithms. If the window is increased, recall that XPLIT, by design, has a waiting time of one RTT_{min} in the worst case. Remember also that if the window is decreased, the working point is also reached in less than RTT_{min} if $\mu_i(t) > \mu_i^{min}$, where μ_i^{min} has been defined in Equation (6.14). In case $\mu_i(t) < \mu_i^{min}$, the queue is stalled and then, we have to invoke the specific procedure

[2] A typical setting for the Best Effort traffic class is to use $\beta_i^{ref} = B_i/2$. In this particular case, we can define a common bound $\Delta\mu_i = \Delta\mu_i^o = \Delta\mu_i^u$.

described in the previous section to handle this situation in which there are no corrective actions to perform until this condition changes.

On the other hand, there are also events produced by a modification of the number of active connections. In this case, the XPLIT manager receives *registration* and *unregistration* events from the *XPLIT sender* processes (see Fig. 6.3). More specifically, a registration event is generated when new data packets arrive from an Internet source to a *XPLIT sender* process. Similarly, an *unregistration* event is generated when the Internet source host is not feeding the *XPLIT sender* process (i.e. the TCP connection ends or it goes idle). The *XPLIT senders* inform the XPLIT manager about these registrations and unregistrations. A signaling protocol is not needed to inform the XPLIT manager since this communication takes place within a unique host. In this case, any Inter-Process Communication (IPC) mechanism can be used. Notice also that registration/unregistration events do not imply changes in the system load but only a different way of sharing this load. In other words, the variation of $N_i(t)$ is related to fairness among active connections. In particular, if we apply a fairness policy in which the system load is evenly distributed among the active flows of a traffic class, we can use Equation (6.5) to update the congestion window of each flow of each traffic class.

Once again, a window update may result in a window increase or in a window decrease, which are not symmetric operations. When there is a new active connection, the windows of the previously active connections will be decreased to accommodate the new one. After applying the window update, XPLIT, as a window control mechanism, will prevent the previously active connections from sending packets until there is enough room for the packets of the new connection. In case there is one active connection less, the window of the active senders have to be increased. However, this action should not be performed immediately because there are still in flight packets of the exiting connection. To be conservative, before increasing the window of the remaining active senders, XPLIT waits enough time to ensure that the share of the leaving connection has left the queue. Given that the packet injection rate of active connections is maintained, the system must wait for $\beta_i^{ref}/\mu_i(t)$. Notice that after this time, there will be room in the queue to immediately accommodate all the additional packets that result of the window increase. Once applied the window update, the periodic management can be activated again after a RTT_{min}.

Finally, we have to study the response timeliness to events of our system. As previously stated, there are two type of events: variations on the available service rate and variations on the number of active connections. Regarding the former type of events, the response timeliness of our system when there is a variation on the number of active connections is almost immediate since this is a local event for the XPLIT manager. Regarding the latter, variations on the available service rate are more critical. This is due to, these events are notified via trap messages, which are not local communications. For this reason, we have to study the signaling strategy when there are events of this kind.

More specifically, the response timeliness in this critical case is determined by the signaling delay T_{sig} between the XPLIT signaler (placed at the Hub) and the XPLIT manager (see Fig. 6.3). During T_{sig} the packet arrival rate λ_i and the service rate μ_i are mismatched and the queue variation is $(\lambda_i - \mu_i)t = \Delta\mu_i t$. While the XPLIT manager is not aware of this situation, we have to avoid two critical situations: buffer overflows (congestion) and buffer underflows (not full link exploitation). Recall that Equations (6.18) and (6.19) defined the critical bounds for the service rate variation that could be managed by the polling mechanism. Following a similar reasoning, we can obtain new critical bounds $(\Delta\mu_i^{u'}, \Delta\mu_i^{o'})$ for the service rate variations that can be managed with our signaling mechanism due to the the signaling delay T_{sig}:

$$\Delta\mu_i^{u'} = \frac{\beta_i^{ref}}{T_{sig}} \tag{6.20}$$

$$\Delta\mu_i^{o'} = \frac{B_i - \beta_i^{ref}}{T_{sig}} \tag{6.21}$$

Let us define α as $T_{sig} = \alpha \cdot RTT_{min}$. In practice, T_{sig} is about few milliseconds, whilst the RTT_{min} in our scenario is about half a second, therefore $\alpha \approx 10^{-2}$. For example, if we use $B_i = BDP_i$ and $\beta_i^{ref} = B_i/2$, then,

$$\Delta\mu_i^{u} = \Delta\mu_i^{o} = \frac{\mu_i}{2} \tag{6.22}$$

and

$$\Delta\mu_i^{u'} = \Delta\mu_i^{o'} = \frac{\mu_i}{2\alpha} \approx 50\mu_i \tag{6.23}$$

As it is observed, the polling mechanism is able to correct slow variations under a 50% of the working service rate, avoiding buffer underflows and overflows. On its side, the trap mechanism can correct quick variations from a 50% to a 5000% of the working service rate, also avoiding buffer underflows and overflows.

6.4 XPLIT performance results (DVB-S2)

In this section, the performance evaluation results of XPLIT architecture is performed. These results have been taken in a variety of conditions, which include short-lived connections, long-lived connections, sets of competing flows, rain events, highly dynamic web traffic and high error rates. For comparison purposes, some of these results are compared with a TCP splitting solution that does not use cross-layer. As it will be shown, the architecture performs extremely well under all these situations, achieving almost the ideal behavior for throughput, user response time and fairness. The results included in this article has been obtained using our NS-2 [DP06] implementation of the XPLIT architecture. However, a more extensive document with more results and a proof-of-concept of the architecture using the NS-2 emulation mode and an external emulator for the DVB-S2 link called AINE [Pa10] is publicly available in the final report of the ARTES project of the European Spatial Agency (ESA) that funded this research (”*IP-friendly cross-layer optimization of adaptive satellite systems*“) [esa07].

Regarding the evaluation criteria, short-lived and long-lived connections are considered. This classification is also used by other authors to validate their proposals [Saa03, OK04]. Short-lived connections are TCP connections that use HTTP to transfer up to some kilobytes. Short-transfers can be further classified in three types depending on the number of objects included in the main web page. On the other hand, long-lived connections are used to transfer large files. A TCP connection is considered long-lived if it has enough duration to complete the slow start phase of a standard TCP and to reach a transmission rate close to the available network bandwidth. To maintain a similar taxonomy, three transfer sizes have also been specified for long-lived connections. Table 6.1 shows these transfer sizes. The number of bytes for each case has been calculated using the MSS previously specified.

Table 6.1: Transfer sizes for Short-lived and Long-lived connections

Label	Short-lived conn.	Long-lived conn.
small	12132 bytes	1 Mbytes
medium	88968 bytes	10 Mbytes
large	148280 bytes	50 Mbytes

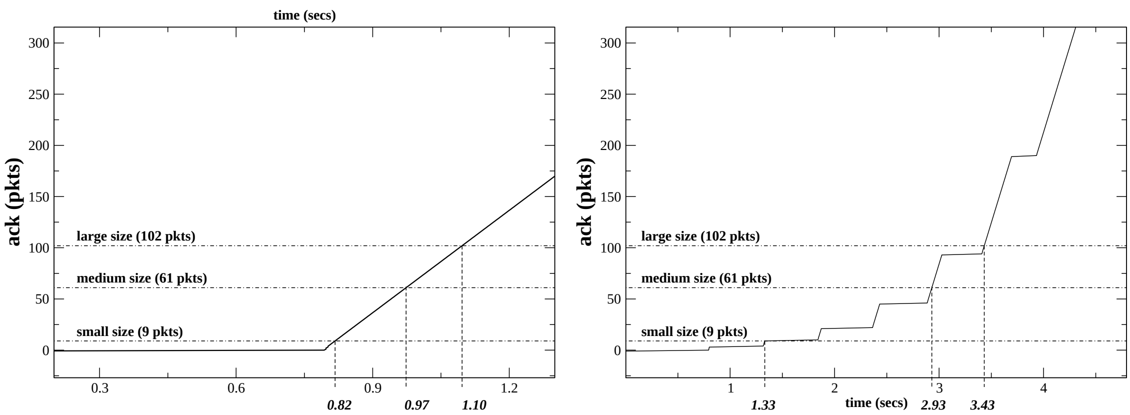
On the other hand, the response time is the metric used to measure the connection performance. This metric is defined as the time that a connection takes to transmit a file. In practice, the transfer is considered finished when the last ACK is received by the HUB.

Finally, it is worth to mention that some of the results presented in this section are compared with a baseline TCP splitting scenario that uses a standard Reno TCP with especial enhancements for satellite networks. These enhancements consider an initial congestion window (IW) of up to 2 segments or 4Kbytes.

Other enhancements added to the TCP baseline for high performance over broadband satellite networks are the window scaling and the time-stamp options proposed in [JBB92]. These enhancements are recommended for best practice in satellite communications. In addition, we also consider the Selective Acknowledgments (SACK) option for the baseline configuration. Using the SACK option, the receiver can inform the sender about arbitrary segments that have been received, regardless of the order in which they arrived. From here on, we will name the previous baseline configuration as *Standard TCP Splitting.*

6.4.1 Short-lived Connections

Fig. 6.5 shows the transfer time evolution for short-lived connections. In the case of XPLIT, as it is shown on Fig. 6.5(a), all connections show a common latency of 1.5 RTT_{min}, which corresponds to the connection establishment. We can also observe that the transfer time increases linearly with the file size. We obtain the link utilization computing the slope of the graph, which is roughly 333 packets/s. This is almost the fully utilization of the link (333.33 packets/s).



(a) XPLIT

(b) Standard TCP Splitting

Figure 6.5: Transfer times for short-lived connections

On the other hand, Fig. 6.5(b) shows the behavior of the baseline *Standard TCP Splitting* for short-lived connections. We can observe that it exhibits the classical slow start phase during all the transfer time, which causes a high response time. The SACK mechanism does not operate because there are no losses for short-lived connections.

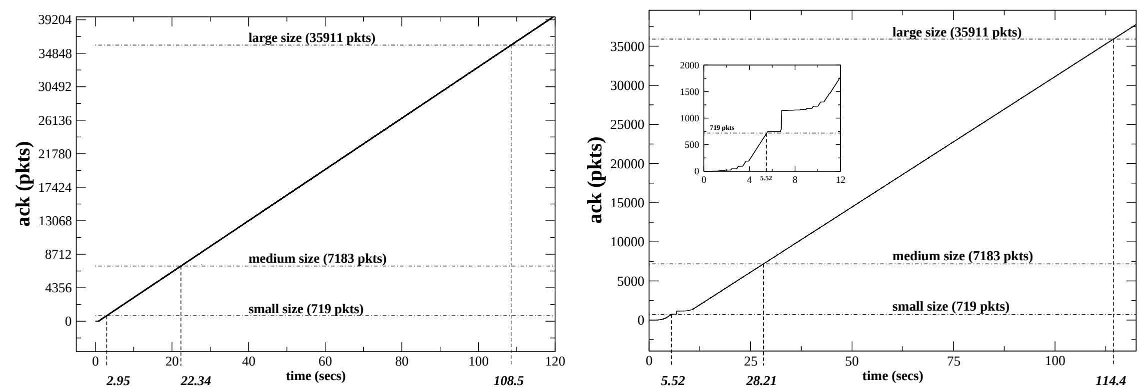
Table 6.2: Transfer time for short-lived connections.

Transfer size (bytes) improv.		**Std TCP Splitting**	**XPLIT arch.**	**% improv.**
Small	12132	0.55 s	0.04 s	92.7
Medium	88968	2.15 s	0.19 s	91.16
Large	148280	2.65 s	0.32 s	87.9

Finally, Table 6.2 shows the comparison between the transfer time for XPLIT and the *Standard TCP Splitting*, and the improvement achieved by XPLIT. Notice that the transfer time improvement is around 90% when using our XPLIT architecture for short-lived connections (we have not considered the common latency of 1.5 RTT_{min} due to connection establishment).

6.4.2 Long-lived Connections

Fig. 6.6 shows the transfer time evolution for long-lived connections for XPLIT and for the baseline *Standard TCP Splitting*. For these simulations, we have used a buffer of 200 packets (roughly BDP), which enables the baseline *Standard TCP Splitting* to use the full link capacity. On the other hand, for comparison purposes, XPLIT is configured to use a reference value of half the buffer size $\beta_{ref} = BDP/2$.



(a) XPLIT (b) Standard TCP Splitting

Figure 6.6: Transfer times for long-lived connections

Fig. 6.6(a) depicts the transfer time evolution for long-lived connections using our XPLIT architecture. As it is shown, the connection establishment is the only delay in the transfer time with respect to the ideal case. For instance, taking one of the cases of Fig. 6.6(a), the transfer time for 719 packets is 2.95 s. If we subtract the 1.5 RTT_{min} connection establishment latency, the transfer time is 2.17 s. In an ideal link with a transmission rate of 333.33 pkts/s, the transfer time for 719 packets is 2.16 s. This result shows the excellent performance obtained in our XPLIT architecture.

We compare these results with the transfer times for long-lived connections of a baseline *Standard TCP Splitting* scenario depicted on Fig. 6.6(b). In this case, transfer times are significantly increased due to the slow start phase of the baseline standard TCP and to the connection establishment latency.

It is worth to mention that the system is configured with a buffer of BDP in order to obtain the maximum performance of the baseline *Standard TCP Splitting* algorithm. When the buffer is empty, a packet suffers a delay of one RTT_{min}, while when the buffer is full the packet suffers a delay of two RTT_{min}, which yields a mean packet latency of 1.5 RTT_{min} (where $RTT_{min} = 0.520$s). Decreasing this buffer size to reduce packet latency will cause the baseline *Standard TCP Splitting* not to fully exploit the link. On the other hand, our XPLIT architecture uses a β_{ref} of half the buffer size, which causes a delay of 1.5 RTT_{min} for all the packets. The XPLIT algorithm can also be configured with a lower β_{ref} to reduce packet latency without link underutilization.

Table 6.3 shows a summary of transfer times for long-lived connections using the *Standard TCP Splitting* and XPLIT, and the improvement achieved by XPLIT. These transfer times have been calculated without considering the common latency of 1.5 RTT_{min}. For long-lived connections that transfer small size files, the transfer time enhancement can reach figures above the 50%. As shown in Table 6.3, the long-term behavior of both architectures (XPLIT and *Standard TCP Splitting*) is similar for large size files. This is because the error probability is approximately null in the DVB-S2 scenario. In this case, the enhancement of XPLIT is mainly due to the fact that we do not have slow start. As

Table 6.3: Long-lived connection transfer times.

Transfer size (bytes)		Std TCP Splitting	XPLIT arch.	% improv.
Small size	1000	4.74 s	2.17 s	54.2
Medium size	10000	27.43 s	21.56 s	21.4
Large size	50000	113.6 s	107.2 s	5.2

a result, the impact of the slow start is less important for large size files. However, the baseline TCP exhibits more latency and a poor performance when there are bandwidth variations, while XPLIT perfectly follows these bandwidth variations.

6.5 Conclusions

In this chapter, a cross-layer PEP architecture is proposed to provide QoS guarantees for TCP flows in the context of the ETSI-BSM-QoS satellite system. The design is focused on the SI layers which is based on a cross-layer optimization. It proposes a cross-layer TCP protocol (called XPLIT-TCP) that uses two control loops to properly manage the system load. The architecture design considers the use of PEPs, as mandatory elements to continuously update the service rate and buffer occupancy values for achieving an efficient and fair bandwidth allocation. The adoption of such architecture allows to guarantee QoS requirements for several DiffServ-TCP flows while keeping the dynamics of the system in control.

The proposed XPLIT-TCP protocol is evaluated in the context of the XPLIT architecture using the NS-2 simulation tool. Simulation results have shown that the proposed XPLIT-TCP variant is able to fairly share available bandwidth over the DVB-S2 forward channel with practically none congestion losses, while having a constant bounded delay per each DiffServ queue. Moreover, it has been identified that both parameters: *the buffer occupancy and the service rate (β_i and μ_i)* are the only required cross-layer parameters to provide optimized congestion control functions.

In the next chapter the design of the XPLIT architecture is developed considering the intrinsic characteristics present for the DVB-RCS return channel.

CHAPTER 7

XPLIT-RCS a cross-layer PEP architecture for the return channel

The Digital Video Broadcasting-Return Channel Satellite (DVB-RCS) link, which represents the bottleneck due to its limited capacity, plays a crucial role in order to provide access to a large number of end-users using the uplink channel. However, one of the most important issues to consider in order to provide interactive multimedia services (such as VoIP) over BSM satellite systems, it is the necessity for supporting on-demand QoS guarantees.

Given that the DVB-RCS link bandwidth is shared among several RCSTs, it is necessary to use Bandwidth on Demand (BoD) techniques controlled by the Network Control Center (NCC) to assign the required amount of bandwidth requested by each RCST. In this way, the DVB-RCS standard [ets05a] proposes the use of the Demand Assignment Multiple Access (DAMA) techniques in order to allow the interactivity supported by different types of services, while enabling the management of the satellite resources efficiently.

In particular, when working with the DAMA scheme, the RCST is responsible for analyzing, estimating and requesting the required capacity for uplink transmissions to the NCC. This is done, each time a RCST requires sending data to a destination host. Here, the NCC is responsible for allocating the return channel time slots based on each request, while informing all the RCSTs about the allowable transmissions slots. This is done using the Terminal Burst Time Plan (TBTP) messages sent over the forward channel. Each RCST receives the allocated TBTP slots and therefore these only transmit data during the allocated slots.

The DVB-RCS standard [ets05a] defines four main capacity allocation types over the return link. Such allocations types can be used to apply different QoS degrees over the return channel. These are supported by the DAMA protocol such as: Continuous Rate Assignment (CRA), Rate-Based Dynamic Capacity (RBDC) Volume Based Dynamic Capacity (VBDC) and Free Capacity Assignment (FCA).

- The CRA class implements a static and constant rate capacity and it is no subject of dynamic request. It is used for the highest priority delay-sensitive user traffic that requires a fixed guaranteed rate, minimum delay and minimum delay jitter.
- The RBDC class is used for high priority variable rate traffic that can tolerate the dynamic response time latency of the dynamic MAC scheduler. It provides dynamic rate assignment (in slots/frames) in response to dynamic request from a ST in order to track the instantaneous traffic rate. It is used for traffic that does not demand a fixed guarantee rate, and supports the minimum scheduling latency delay.
- The VBDC class is used for traffic that can tolerate delay and jitter effects (i.e. BE traffic). VBDC capacity is provided in response to dynamic volume request from each Return Channel Satellite Terminal (RCST) to the MAC scheduler. The VBDC capacity is not guaranteed as it is assigned as BE capacity within the available resources, after satisfying the total CRA and RBDC.
- Finally, the FCA class uses the capacity that has not been allocated from one or more RCST. This assignment is not explicitly requested by the RCST because, the capacity assignment to the FCA class comes from the capacity which would be otherwise unused.

In particular, when dealing with the provisioning of QoS guarantees, the CRA and RBDC types can offer different guarantees, while the VBDC and FCA types represent the best effort services. In this way, either the RBDC or the VDBC are based on capacity requests, while the CRA and FCA types do not involve capacity requests and therefore may reduce the delay.

In this chapter, an extension of the XPLIT architecture called XPLIT-RCS, developed for the return channel is presented. This represents a complementary configuration working together with the forward channel (previously described on Chapter 6). This return channel design is also based on PEPs and cross-layer techniques. Hence, the XPLIT-RCS architecture design considers the DAMA techniques on the return link to manage the bandwidth assignment required by each RCST terminal to meet its demands.

In this scenario, it is important to bear in mind the effects that the DAMA techniques may introduce [LRF09], [GPS06]. As it may generate variable RTTs and higher propagation delays, consequently adding strong and abrupt variations in the available bandwidth [ipe05]. In addition, one of the main concerns when using DAMA protocol is the delay accumulation [GWE+09] that can severely affect the performance of the TCP protocol.

The XPLIT-RCS architecture is developed based on a collaborative design between the *PEP* and the RCST terminal. In order to improve the performance of the return satellite channel, the XPLIT-TCP variant (previously described on Section 6.3) has been selected as a satellite-optimized TCP protocol that uses cross-layer information reported by lower layers to improve the performance of TCP flows in the satellite segment.

The XPLIT-RCS architecture design is developed in compliance with the ETSI-BSM-QoS [10205], the standard DVB-RCS [ets05a] and the I-PEP specification defined by the

SatLabs working group [Sat08] in order to enhance TCP transmissions, while proving QoS guarantees over the return satellite channel.

In this way, the I-PEP specification defined by the SatLabs working group [Sat08], specifies a concept called *Interoperable PEP* that defines the basic procedures and protocol messages exchanged over the air interface between the satellite terminal and the Hub station. This is done in order to adapt a suitable transmission protocol used for satellite communications via the DVB-RCS link, while maintaining backward compatibility with standard TCP variants and the Space Communications Protocol Specifications- Transport Protocol (SCPS-TP) [SBB06].

7.1 XPLIT-RCS scenario

The XPLIT-RCS architecture is developed based on a collaborative design between the *PEP* and the RCST terminal. In particular, both elements are set up inside the Customer Premises Equipment (CPE), which is allocated at the terrestrial segment. This is done in order to allow satellite operators to easily adopt the proposed architecture and algorithms with low deployment cost.

The topology defined for the development of the XPLIT-RCS architecture design is shown in Fig. 7.1.

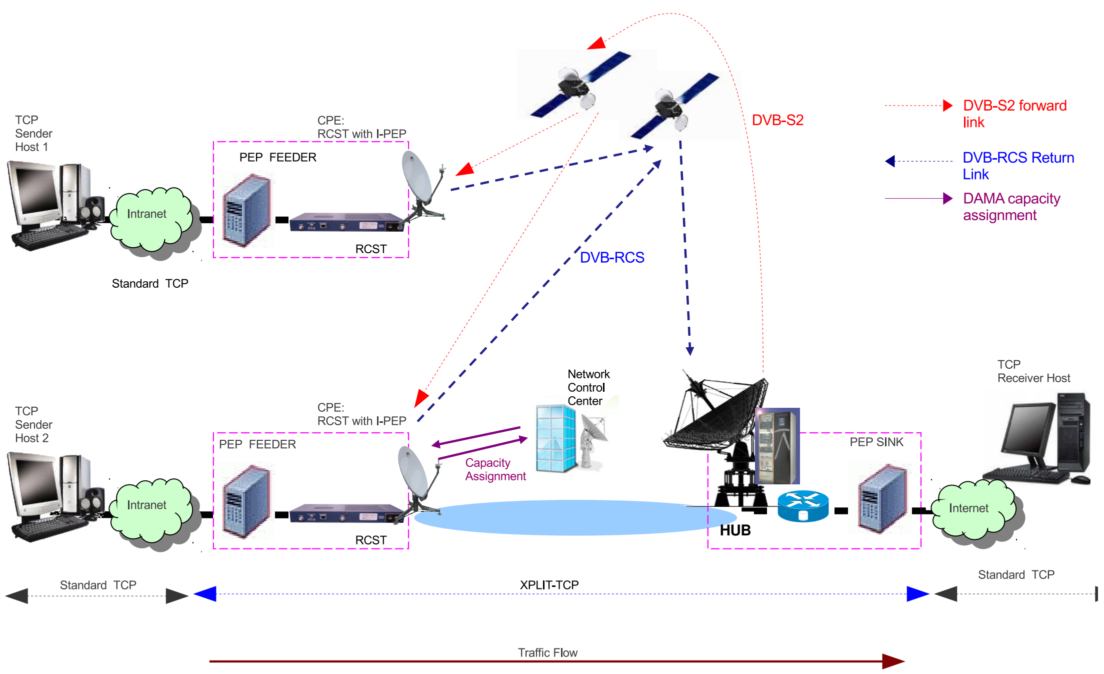


Figure 7.1: Topology for the XPLIT-RCS architecture

This topology considers two GEO satellite equipments in the Ka band (30/20 GHz), working in a transparent multi-beam architecture. Here, two remote sources send data to a destination allocated at the Internet side by means of the return satellite channel. For clarity purposes, the DVB-S2 forward links are also depicted on Fig. 7.1.

In the subscriber-side, each RCST is defined as a mandatory element for managing the satellite Mac/Phy layers. Similarly, the *PEP feeder*, represented by a CPE-PEP box, is defined to manage and control the parameters to efficiently manage the TCP transmission at the return link.

The RCST is responsible for sending the data to the Hub by means of the DVB-RCS return satellite channel. Each data transmission preformed by the RCST is controlled by the NCC, with the support of the BoD mechanism based on the the DAMA protocol, to share the radio spectrum among the remote users.

Here, the NCC plays a key role for the operational and managements tasks performed by the satellite operator, which allows to manage the functional parameters to establish priority levels and traffic rates according to the defined Service Level Agreements (SLAs). Such specifications, are set up each time a RCST terminal is logged into the Hub for guaranteeing the different traffic classes over the return link.

Moreover, the assignment of resources to each satellite terminal is dynamically controlled by the NCC. This is done based on the capacity requests made to the NCC and the boundary values negotiated during the connection establishment. These assignments are conditioned by the availability of resources at the return channel. On Fig. 7.1, the transmitted data arrives at the Hub which is responsible for managing the TCP connections using the *PEP sink*, while routing the data through the Internet cloud to the destination terminal.

7.2 XPLIT-RCS architecture functional blocks

The XPLIT-RCS architecture design is depicted on Fig. 7.2. As it is shown, incoming flows are received by the *PEP feeder* to be forwarded through the RCST to the destination host.

In particular, this PEP uses *TCP feeders* to allow interoperability among standard TCP versions and the XPLIT-TCP protocol (described in subsection 6.3). The physical connection between the RCST and the PEP is performed by means of a Local Area Network (LAN).

With the aim of guaranteeing the Interactive Satellite Network (ISN) operation, the satellite resources utilization is managed using Bandwidth on Demand (BoD) schemes between the RCST and the NCC. These BoD functions are allocated at the Satellite Media Access Control (SMAC) layer, as it is shown on Fig 7.2.

The BoD scheme considers a set of MAC protocols and algorithms that allow each RCST requests the NCC to provide the satellite resources, every time the transport of traffic is needed. Here, the use of DAMA techniques are adopted to improve the utilization of satellite resources when different traffic classes are present.

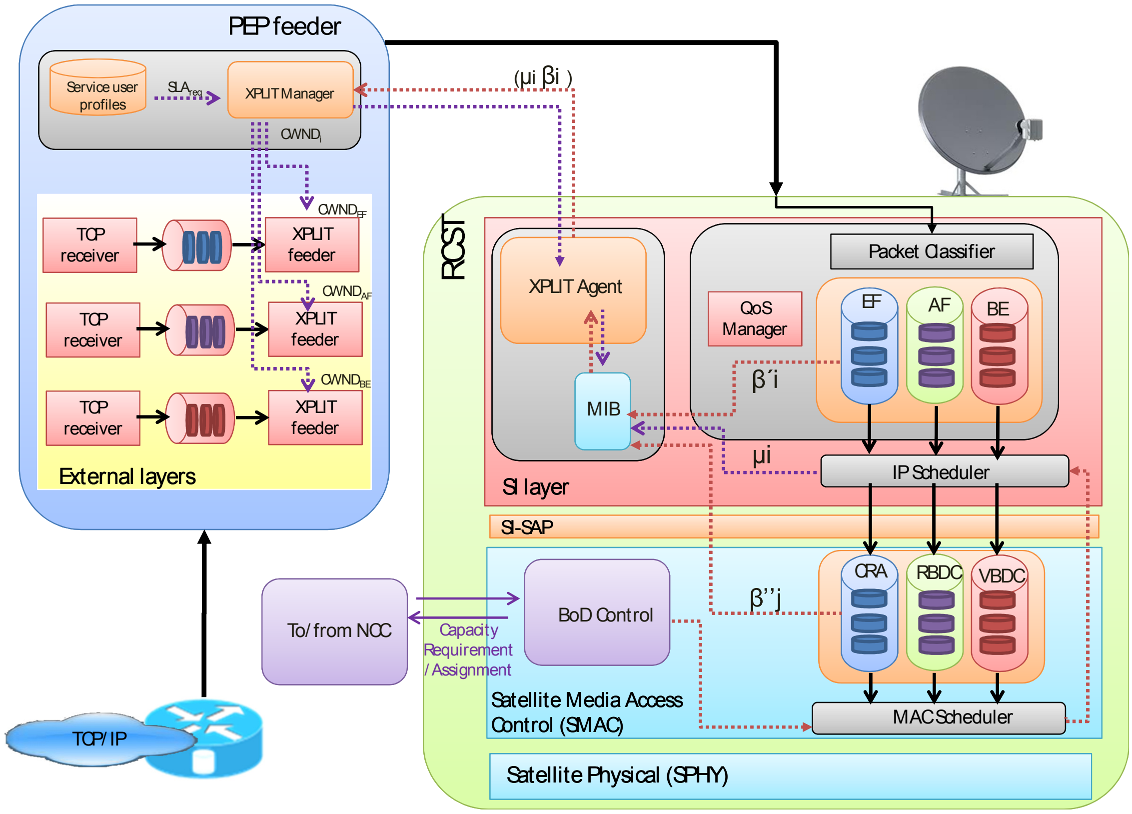


Figure 7.2: XPLIT-RCS architecture design

Inside the RCST (at the SMAC layer), the QoS blocks and their functionalities are defined, as a part of the proposed cross-layer design (see Fig 7.2). At the SI layer, the packet classifier, the QoS manager, the set of DiffServ queues and the IP scheduler are set up. At the SD layer, the DAMA buffers are allocated considering three queues supporting different DAMA assignment types. Here, packets wait to be forwarded to the destination host using the MAC scheduler.

Without lost of generality, at the SI layers, the set of DiffServ traffic classes have been reduced to three which are related to the EF, AF and BE DiffServ traffic classes. Similarly, at the SMAC layer, three physical queues are set up, which are associated with the RCA, RBDC and VBDC capacity types.

It is worth mentioning that the XPLIT-RCS architecture design complies with the ETSI-BSM-QoS functional architecture supported by the standards ETSI TS 102 157 [ETS03] and ETSI TS 102 462 [10205].

7.2.1 The QoS mapping design

In order to match the DiffServ forwarding requirements at the SI layers, the QoS traffic classes are appropriately mapped into the DAMA capacity types supported by the MAC scheduler.

The QoS mapping between DiffServ classes and DVB-RCS capacity assignments types are defined following the guidelines provided in [M M08] and [Sat08]. In general higher priority service classes are associated with guaranteed capacity such as the CRA or RBDC types, while lower priority capacity are predominately given by the BE capacity such as the VBDC and FCA types.

In particular, the design of the XPLIT-RCS architecture considers the QoS requirements (delay characteristics) in order to define the mapping of the MAC capacity assignment into the DiffServ capacity types. Therefore, the EF traffic class is mapped into the CRA capacity assignment, the AF traffic class is mapped into the RBDC capacity type and the BE traffic class is mapped into the VBDC capacity type. The QoS Mapping between the DiffServ traffic classes and the DAMA capacity assignments, considering the delay constraint characteristics are specified on Table 7.1.

Table 7.1: QoS Mapping between DiffServ traffic classes and DAMA capacity assignments

IP App	Acceptable Delay	DiffServ class	DAMA type
VoIP	300 ms	EF	CRA
HTTP or Video server	850 ms	AF	RBDC
FTP server	not controlled	BE	VBDC

7.2.2 The queue design

The design showed on Fig 7.2 depicts the QoS buffers and the DAMA buffers separately, which are allocated at the SI and the SMAC layers respectively. Nevertheless, in most of the cases, the CPE is configured considering joint queues, allowing to have one queue for each traffic class.

In this context, to fully exploit the satellite link [LM97], [MSMO97] at the return channel, the buffer sizes are set up considering:

- That the first packet in the system (considering that it is empty) takes $2 \times RTT_{min}$ to be confirmed at the sender side by receiving its corresponding ACK message. This value includes the time a DAMA capacity assignment takes place (which is set to $1 \times RTT_{min}$), given that the first packet requires to be assigned with the capacity provided by the NCC. Here, RTT_{min} represents the two-way propagation delay or minimum Round Trip Time (set to 560 ms in the GEO satellite scenario).

- Therefore, to keep the transmission of packets constantly during this period of time ($2 \times RTT_{min}$), it is necessary to guarantee a minimum load given by $Load_{min} = 2 \times BDP$. Here, BDP represents the satellite *Bandwidth Delay Product.*

- In this way, the minimum TCP *cwnd* should reach the maximum load, set to $2 \times Load_{min} = 4 \times BDP$. This is done in order to guarantee the continuity of the transmission, being able to detect a packet lost, as thus reducing its *cwnd* value to the half ($Load_{min}$).

- As a result, the minimum XPLIT buffer size (B_{min_i}) for each traffic class is set to:

$$B_{min_i} = 3 \times \mu_i \cdot RTT_{min} = 3 \times BDP \tag{7.1}$$

Here, B_{min_i} represents the minimum buffer size (defined in packets) and μ_i represents the rates for each traffic class i.

In order to measure the buffer occupancy (β_i) level for each traffic class, the XPLIT-RCS design considers two queue occupancy values. The former is measured at the SMAC layer while the later is assessed at the SI layer (see Fig. 7.2). Therefore, total buffer occupancy (β_i) at the CPE equipment for each traffic class i is defined as follows:

$$\beta_i = \beta_i' + \beta_j'' \tag{7.2}$$

Here it is worth mentioning that the element called *QoS manager* (at the RCST) monitors the occupancy of the different queues β_i' and β_j'' (see Fig 7.2) in order to know the total occupancy of the system.

7.2.3 The cross-layer design

The XPLIT-RCS architecture considers a cross-layer design to share information between the RCST and the PEP in order to improve TCP performance over the return satellite channel.

In particular, the transport protocol used to send data through the return channel, is the XPLIT-TCP variant [AMDMn+12], which was previously specified on Section 6.3. This protocol requires two main cross-layer parameters set up at each RCST: the buffer occupancy and the service rate (β_i and μ_i), as it is shown in Fig. 7.2.

Here, every time the RCST receives the resource allocation (service rate (μ_i for each DAMA capacity type) by means of a TBTP message from the NCC, the service rate (μ_i) is assessed at the IP scheduler from the the SI layer. Both parameters (μ_i and β_i) are forwarded to the *PEP feeder* using a cross-layer mechanism. Here, the transport layer is responsible for receiving the cross-layer parameters which are used to modify and adjust the *cwnd* values accordingly for each TCP transmission. The cross-layer design and the exchange of the parameters β_i and μ_i are also shown on Fig 7.2.

Following the forward channel design (Section 6.3), XPLIT-TCP uses two control loops to manage the congestion changes in the satellite system. The first control loop uses periodic polling of the cross-layer information μ_i and β_i to regulate the slow dynamics of the system (i.e. new TCP connections) and a second control loop to manage the fast and sudden dynamics which are mainly due to changes of available bandwidth assignment performed by the NCC and the DAMA scheme.

Similar to the XPLIT-FWD architecture design (Section 6.2.1), the use of SNMP protocol is adopted to provide signaling between the PEP and the RCST. This is done, using the *XPLIT Manager* and the *XPLIT Agent* modules, as it is observed on Fig 7.2.

In particular, the *XPLIT manager* (allocated at the PEP) initiates the XPLIT-TCP feeders and manages them during the connection duration. The *XPLIT manager* is able to know the DAMA capacity assignments measured at each time, while adjusting the *cwnd window size* for each active XPLIT-TCP connection. This is done, considering the service rate and buffer occupancy values (μ_i and β_i) reported by each RCST. Here, it is worth mentioning that the sharing of resources among XPLIT connections is performed considering the fair distribution of satellite resources.

On the other hand, each RCST uses the *XPLIT agent* to manage the cross-layer information parameters reported from the IP scheduler (μ_i) and the selected queues (β'_i and β''_j). In this way, the service rate μ_i is computed by the *IP scheduler*, which controls the number of packets for each traffic class. The service rate μ_i considers the capacity assigned by the NCC. This value is reported by the MAC scheduler using a second cross-layer mechanism defined between the SMAC layer and the SI layer (see Fig 7.2).

The cross-layer parameters are stored in the *XPLIT MIB* (Management Information Base), which are fed back to the *XPLIT manager* by means of the SNMP primitives.

It is worth mentioning that in [RMAMD+13] and [RMMDA+09], the complete design of the XPLIT-RCS architecture is developed, together with an extensive performance evaluation of the XPLIT-TCP variant.

7.3 XPLIT-RCS performance evaluation

In this subsection, the performance evaluation for XPLIT RCS architecture is carry out using the NS-2 simulation tool version 2.29. Here, the XPLIT-RCS architecture is simulated following the topology defined in Fig. 7.1. It considers two remote sources, sending two flows (each one) to a destination allocated at the Internet side by means of the return GEO satellite link.

Importantly, the presented simulation scenario only considers the evaluation of the VBDC traffic type. This is done to evaluate the XPLIT-TCP variant working in a controlled environment, in which the dynamic assignment of satellite resources are implemented, including the effects that the DAMA techniques introduce, such as variable RTTs and bandwidth variations. Nevertheless, it is worth mentioning that the present analysis could be extended to manage others DAMA capacity types such as the RBDC type.

The main objective to perform the proposed simulation tests is to compare the XPLIT-TCP variant against a baseline Sack TCP, when working in a DVB-RCS environment with the adoption of the DAMA techniques. The presented simulation results are developed considering three tests:

- *Short-lived connections* in which a short file is sent over a single RCST.
- *Long-lived connections* in which a long file is forwarded over a single RCST.
- *Fairness Analysis* in which either two or four files are sent using two RCSTs terminals.

The transfer sizes for Short-lived and Long-lived connections are shown in Table 7.2.

Table 7.2: Transfer sizes for Short-lived and Long-lived connections

Label	Short-lived conn.	Long-lived conn.
file size	1024 bytes	100 Kbytes

Finally, the performance metrics used to evaluate the XPLIT-RCS architecture are: goodput, queue occupancy, and fairness (described in section 4.2.3). The TCP attributes such as *cwnd*, ACKs, RTT and responsiveness are also evaluated. It is worth mentioning that the goodput values calculated during the simulation tests are set considering the approximation $cwnd/RTT$. This is done to add simplicity in the analysis of the obtained results.

7.3.1 Satellite network settings

The satellite return channel capacity C_{rcst} representing the bottleneck link, is set considering a reduced and limited capacity up to 256 Kbps. The TCP variant defined to transport the VBDC capacity type is the baseline Sack TCP [BAFW03], in comparison with the proposed XPLIT-TCP variant. Here both TCP variants consider a buffer size set to 3 BDP (see analysis presented in Section 7.2.2) this is done in order to compare them in similar conditions.

In addition, the TCP Linux version [DP06] is adopted for simulation purposes, which includes the Sack TCP variant among others. Here, several modules programed in TCL were added to perform the XPLIT-RCS functionalities while simulating the DAMA capacity assignment and the XPLIT-TCP protocol.

The DVB-RCS satellite network settings throughout the simulation are configured as it is defined in Table 7.3.

Table 7.3: Simulation system parameters

Parameter	Value	Parameter	Value
RTT_{min}	520 ms	BDP (pkts)	11
RCSTs number	2	Flows x RCST (maximum)	2
Traffic type	VBDC	MF-TDMA frame length	6.625 ms
MF-TDMA super frame length	26.5 ms	Assignment cycle time	63 ms
Time slot x Frame	32	IP MTU size	1500 bytes
Frames x Super Frame	4	Capacity per slot	16 Kbps
RCST log on capacity	32 Kbps	XPLIT Buffer ref	3BDP/2
RCST capacity (C_{rcst})	256 Kbps	Queue size (pkts)	33 (3BDP)
PER	10^{-4}		

7.3.2 Simulation results: Short-lived connections

In this test, a scenario in which a single RCST terminal sends a small file (1024 bytes) through the DVB-RCS link is evaluated. The simulation results of goodput and queue occupancy (short-lived connections) for the transmission of a single VBDC flow are shown on Fig. 7.3. Here, the comparison is performed using either XPLIT-TCP or Sack TCP.

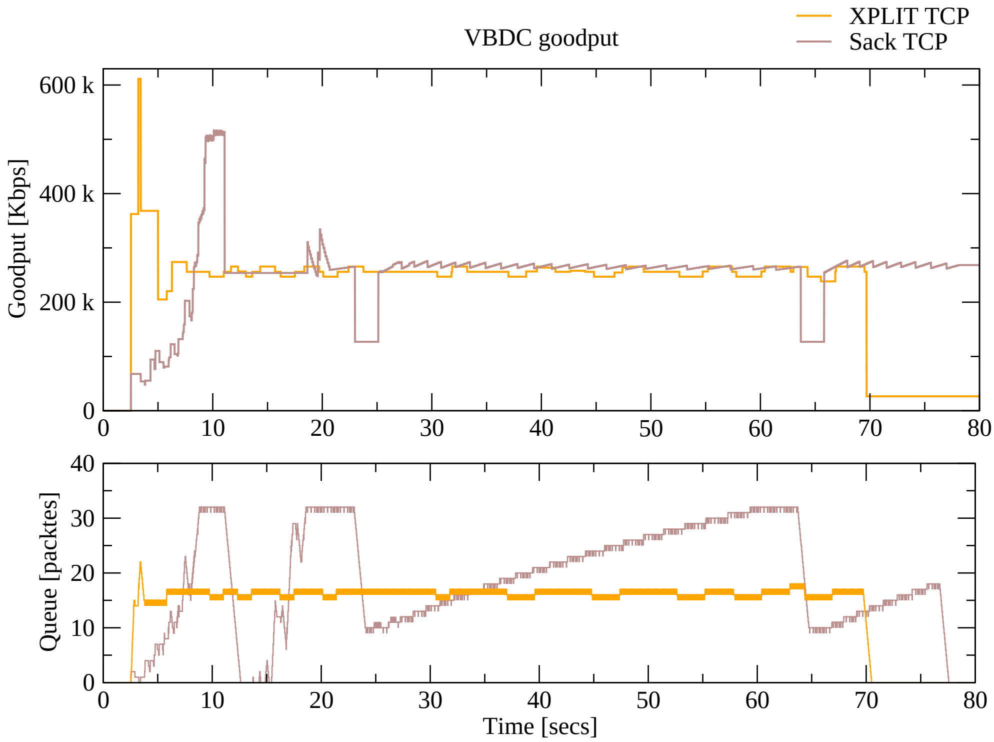


Figure 7.3: Simulation results: Short-lived connection XPLIT vs. Sack (Goodput and Queue occupancy)

As it is observed on Fig. 7.3 (upper graph), using the Sack TCP variant, the goodput level severely fluctuates. Particularly, at the 11-seconds time-point, the VBDC capacity type is able to reach a peak value up to 510 Kbps of goodput. After this point the goodput gets a quasi-constant behavior, being able to reach the assigned RCST capacity set to 256 Kbps. However, at the 24 seconds time-point (also repeated at the 64 seconds time-point), the goodput level drops down the half of the assigned RCST capacity (128 Kbps). This is mainly because the Sack TCP variant is not able to transmit the amount of packets that are filling the queue. Hence, the queue occupancy is overloaded, as it is see in the lower graph, reaching its limit value (set to 33 packets). This situation generates a reduction in the congestion window at these time-points, leading to an increase of the system latency.

In contrast to these results, when using the proposed XPLIT-TCP variant, the VBDC capacity type is able to provide a more constant goodput behavior compared to the previous case, being able to properly use the DAMA BW assignment (set to 256 Kbps). Here, a goodput level up to 256 Kbps is reached during all the simulation (see upper graph). In addition, the queue occupancy is reduced compared with Sack TCP, being able to reach 16 packets constantly. This is a suitable result when working with DAMA DVB-RCS satellite systems, as it is possible to control the satellite system load, while using efficiently the capacity assigned by the NCC.

It is worth mentioning that the XPLIT-TCP responsiveness (measured during the initial TCP phases) reaches the stable value up to 256 Kbps after 5.58 seconds (see Fig. 7.3 upper graph). In contrast to Sack TCP that reaches the same steady state phase after 11 seconds.

In order to complement the first set of tests using only one VBDC flow, Fig 7.4 and Fig 7.5 show the simulation results of *cwnd*, *ack* received packets and the *RTT values*, when evaluating the XPLIT-RCS architecture jointly with the XPLIT-TCP variant in comparison with Sack TCP.

Particularly, Fig. 7.4 (upper graph) illustrates the congestion window (*cwnd)* evolution considering one flow using either Sack TCP or XPLIT TCP. Here, when using Sack TCP,

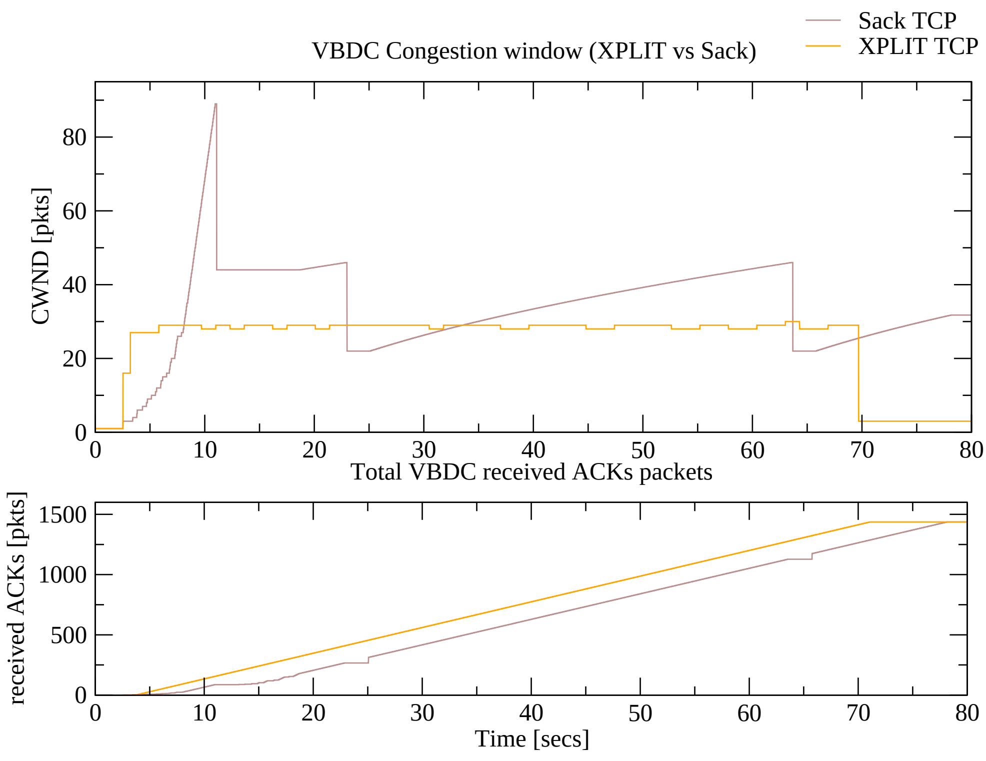


Figure 7.4: Simulation results XPLIT vs Sack (*cwnd* and received ack packets)

it is possible to note that the curve has a plateau between 12 and 24 seconds time-points reaching *cwnd* values around 42 packets. After this point, the *cwnd* drops down to the half of its value, which is mainly because the last packet loss event has occurred. Here, Sack TCP starts the *fast retransmit* and *fast recovery* phases aiming to fill into the limit the queues again. However, at the 64 seconds time-point the *cwnd* value drops again.

In contrast to these results, when XPLIT-TCP is used, the *cwnd* curve remains constant, as it is shown on Fig. 7.4 (upper graph). As a result, the *cwnd* level reaches 30 packets on average during all the simulation.

On the other hand, Fig. 7.4 (lower graph) illustrates the total received VBDC ACK packets. As it is shown, when using the proposed XPLIT-TCP, the maximum number of received ACK packets (approx. 1480) are correctly received at the 69 seconds time-point. Hence, the XPLIT-TCP is able to transmit a complete short file (1024 Kbps) during 69 seconds. However, when using the Sack TCP variant, the same number of received ACK packets (approx. 1480) are reached at the 79 seconds time-point. This measured time is greater compared to the results obtained with XPLIT-TCP which is reduced by 12.66%. As a natural consequence, the less the period of time in which the VBDC packets are received, the faster the TCP transmission is ended. The transfer time for short-lived connections are shown in Table 7.6.

Table 7.4: Transfer time for short-lived connections.

Transfer size (bytes)		**Std TCP Splitting**	**XPLIT arch.**	**% improv.**
Small	1024	79 s	69 s	12.66

In particular, at the 24 and 64 seconds time-points, the *fast retransmit* and *fast recovery* phases for the Sack TCP variant take place. These are represented as an abrupt change in the number of received ACK packets, as it is shown on Fig. 7.4 (lower graph).

Finally, the simulation results concerning the round trip time (RTT) values are shown on Fig. 7.5.

These values are measured considering the mean packets that are received during the simulation time. As it is shown, when using the proposed XPLIT-TCP variant, constant RTT values up to 1.3 seconds are presented. However, when using the Sack TCP variant, the RTT values are increased reaching peak values up to 2.08 seconds. This measured RTT value is greater compared to the results obtained with XPLIT-TCP.

7.3.3 Simulation results: Long-lived connections

In this test, a scenario in which a single RCST terminal sends a long file (100 Kbytes) through the DVB-RCS link is evaluated. The simulation results of goodput and queue occupancy for long-lived connections are shown in Fig. 7.6. These results are performed when evaluating the XPLIT-TCP variant in comparison with Sack TCP.

As it is observed on Fig. 7.6 (upper graph), using the Sack TCP variant, the goodput level severely fluctuates. Although it remains constant during several periods of time

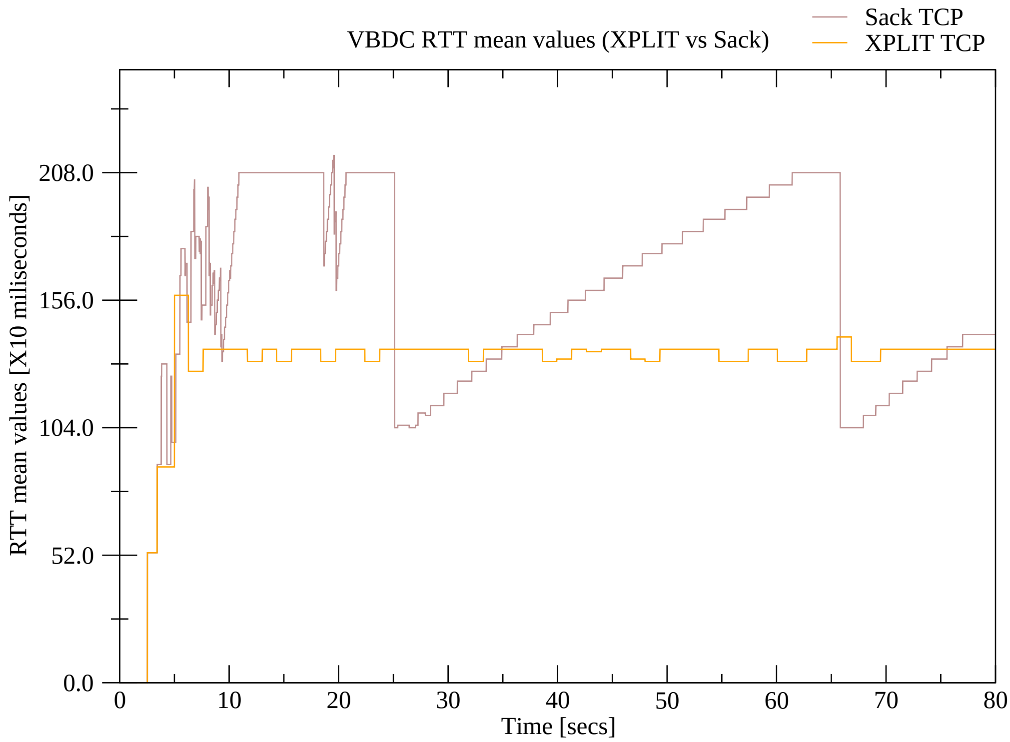


Figure 7.5: Simulation results XPLIT vs Sack (RTT values)

(being able to reach the assigned RCST capacity set to 256 Kbps), in specific time-points (24, 56, 105, 145, and 187) several goodput reductions are experienced (128 Kbps). As it is observed in the lower graph, the queue occupancy shows the typical *sawtooth* behavior, being able to fill the queues up to its maximum value (33 packets) This situation generates a reduction in the congestion window at these time-points, leading to an increase of the system latency.

In contrast to these results, when using the proposed XPLIT-TCP variant, the VBDC capacity type is able to provide a more constant goodput behavior compared to the previous case, in which none drops are experienced (being able to reach 256 Kbps constantly). As a result, the XPLIT-TCP goodput is a able to properly use the DAMA BW assignment (set to 256 Kbps). In addition, the queue occupancy is reduced compared with Sack TCP, being able to reach 16 packets constantly. This is a suitable result when working with DAMA DVB-RCS satellite systems.

In order to complement the long-lived tests, Fig 7.7 shows the simulation results of *cwnd* and the *ack* received packets, when evaluating the XPLIT-RCS architecture jointly with the XPLIT-TCP variant in comparison with Sack TCP.

On Fig 7.7 upper graph, when using Sack TCP, it is possible to see the typical Sack TCP *sawtooth* behavior during all the simulation. In contrast to these results, when

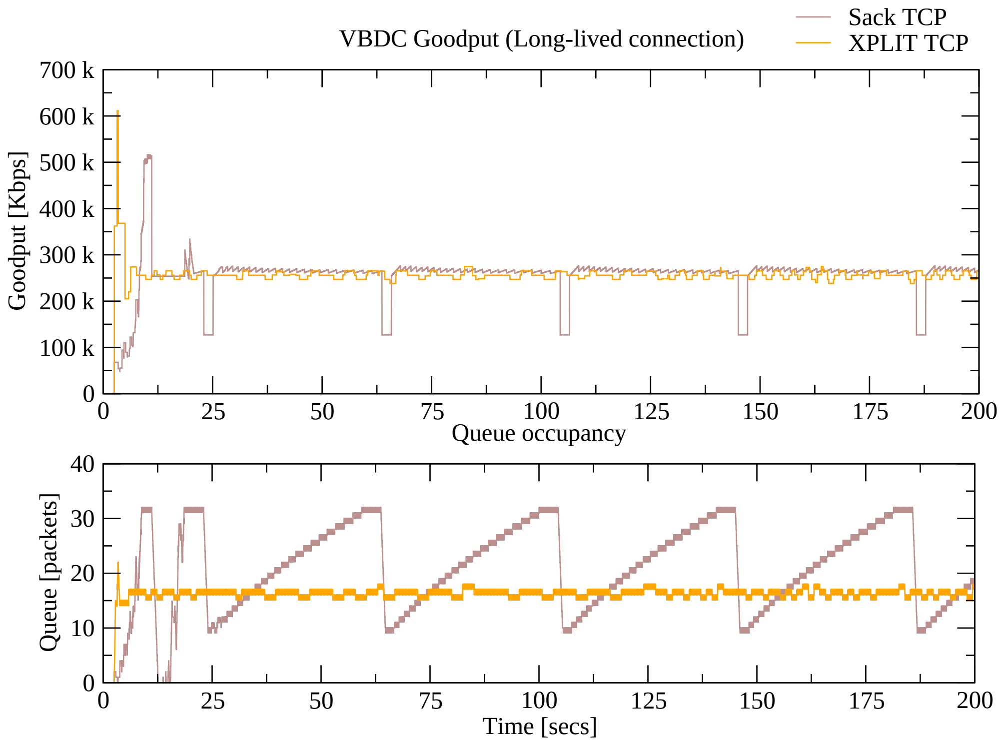


Figure 7.6: Simulation results: Long-lived connection - XPLIT vs. Sack (Goodput and Queue occupancy)

XPLIT-TCP is used, the *cwnd* curve remains constant, as it is shown on Fig. 7.7 (upper graph). As a result, the *cwnd* level reaches 30 packets on average during all the simulation.

On the other hand, Fig. 7.4 (lower graph) illustrates the total received VBDC ACK packets. Here, the similar behaviour analized for short-lived connection is shown (see section 7.3.2). However, the XPLIT-TCP is able to transmit a complete long file (100 Kbps) during 210 seconds. Contrary to Sack TCP variant, which is able to transmit a complete long file at the 270 seconds time-point. This measured time is greater compared to the results obtained with XPLIT-TCP which is reduced by 2.2%.

The transfer time for long-lived connections are shown in Table 7.5.

Table 7.5: Transfer time for long-lived connections.

Transfer size (bytes)		**Std TCP Splitting**	**XPLIT arch.**	**% improv.**
long file	100 Kbps	3348 s	3274 s	2.2

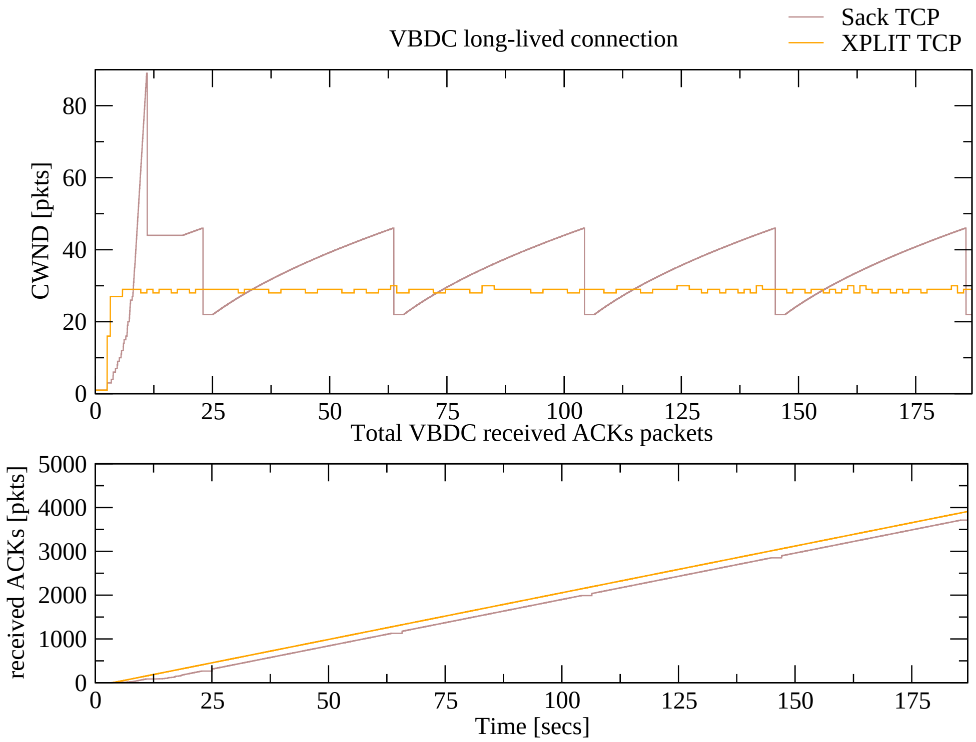


Figure 7.7: Simulation results: Long-lived connection - XPLIT vs Sack (*cwnd* and received ack packets)

7.3.4 Fairness results

The second experiment is proposed in order to evaluate the fairness level reached by each TCP variant in the proposed scenario. Here, a RCST terminal shares the available bandwidth between two TCP flows using either XPLIT-TCP or Sack TCP. The file size is set to 3072 bytes. The DAMA bandwidth assignment is set to 256 Kbps for the single RCST.

The simulation results of goodput and queue occupancy for the VBDC capacity type are shown on Fig. 7.8.

Here, the DAMA bandwidth assignment set to 256 Kbps is also depicted for the single RCST. As it is observed on Fig. 7.8 (lower graph), using the Sack TCP variant, the goodput level for both TCP flows severely fluctuates. As a result, both flows show unacceptable fairness level, in which the flow 2 uses more bandwidth, while penalizing flow 1. Similarly, the queue occupancy is shown on Fig. 7.9, as it is observed when using Sack TCP variant the queue occupancy is overloaded, reaching its limit value (set to 33 packets). Here, the typical *sawtooth* behavior is also shown representing the speed in which the buffer occupancy is filled with packets. However, after several lost of packets, the queue occupancy level is fallen sharply.

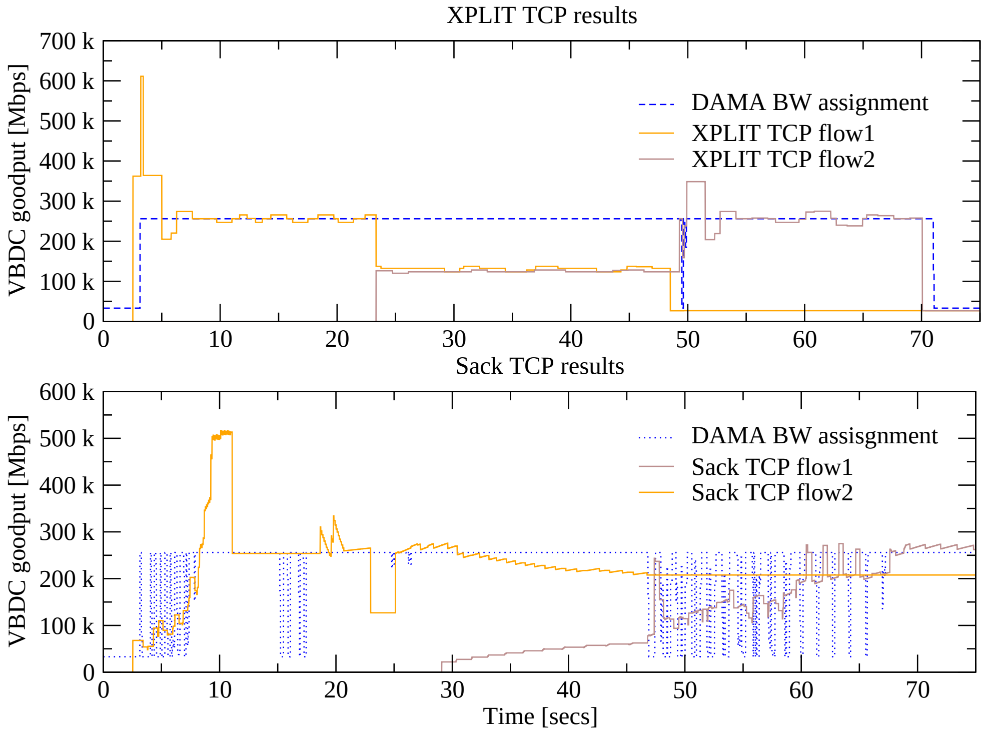


Figure 7.8: Simulation results (two flows) XPLIT vs Sack (Goodput)

In contrast to these results, when using the proposed XPLIT-TCP variant (see Fig. 7.8 -upper graph), both VBDC flows provide a quasi-constant goodput level, being able to fairly share the available bandwidth, while matching the DAMA BW assignment curve. Here, a goodput level up to 125 Kbps is reached during all the simulation. In addition, the queue occupancy is reduced compared with Sack TCP, being able to reach up to 17 packets on average constantly. This is a suitable result when working with DAMA DVB-RCS satellite systems, as it is possible to optimized the satellite system load, while fairly share the available resources.

Finally, the last experiment is proposed in order to analyze the XPLIT-TCP behavior when several TCP flows share the available bandwidth. Here, two RCST terminals send two TCP flows simultaneously. The file size is set to 3072 bytes and the DAMA bandwidth assignment is set to 256 Kbps for the single RCST.

The simulation results of goodput and queue occupancy for the four VBDC flows (using XPLIT TCP) are shown on Fig. 7.10.

It considers that the RCST1 requests the capacity assignment at 2 second time-point, while the RCST2 requests its capacity at 18 seconds time-point.

As it is observed, when using the proposed XPLIT-TCP variant, the four VBDC flows provide quasi-constant goodput behavior, being able to fairly share the available

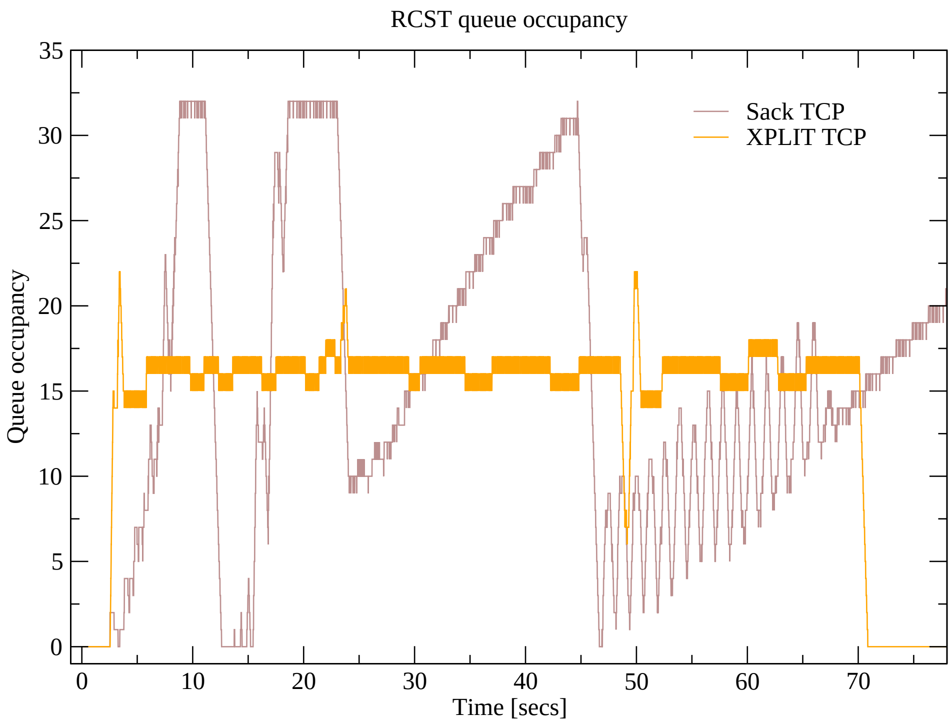


Figure 7.9: Simulation results (two flows) XPLIT vs Sack (Queue occupancy)

Table 7.6: Transfer time for the first TCP connection (four flows).

Transfer size (bytes)		Std TCP Splitting	XPLIT arch.	% improv.
file	3072	299.99 s	150 s	50.1

bandwidth among flows, while matching the DAMA BW assignment curve (see Fig. 7.10 upper graph). Here, constant goodput levels up to 62 Kbps are reached during all the simulation. In addition, the queue occupancy for both RCSTs are constant, being able to reach up to 17 packets. This is a suitable result when working with DAMA DVB-RCS satellite systems, as it is possible to perfectly match the satellite system load, while being able to fairly share the available resources.

In contrast to these results Fig. 7.11 (upper graph) shows the simulation results of goodput and queue occupancy for the four VBDC flows (using Sack TCP). As it is observed the four flows are severely penalized, being unable to reach an acceptable goodput level. As a result, the flows are constantly competing for the satellite system resources showing an unacceptable fairness level. The transmission duration is longer compared with XPLIT

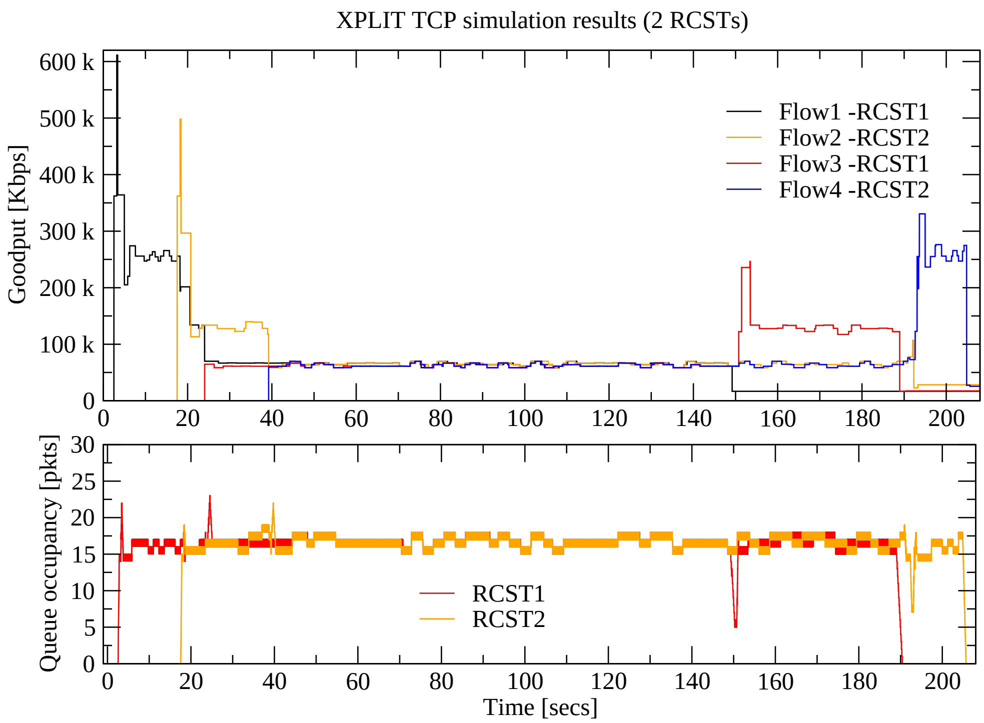


Figure 7.10: Simulation results (four flows) only XPLIT TCP (Goodput and Queue occupancy)

TCP. Similarly, the queue occupancy is shown on Fig. 7.11 (lower graph), as it is observed when using Sack TCP variant the queue occupancy is overloaded, reaching its limit value which is set to 33 packets. Here, the typical *sawtooth* behavior is shown again. However, it is worth mentioning that it is possible to reach the convergence among flows when evaluating the proposed scenario in the the long term simulation, as it is shown on Fig. 7.12.

7.4 Conclusions

In this chapter, the design of the XPLIT-RCS architecture, developed for the return satellite channel, has been presented. It has been designed to provide QoS guarantees over the DVB-RCS/ETSI-BSM QoS satellite systems considering the use of cross-layer techniques together with the implementation of PEPs. The propose architecture also adopts the XPLIT-TCP variant (specified for the forward channel), which has been proven to provide enhanced results when working with the DAMA capacity assignments.

The XPLIT-RCS architecture complies with design goals also defined for the DVB-S2 forward channel, as these are: the source and destination hosts are able to use standard

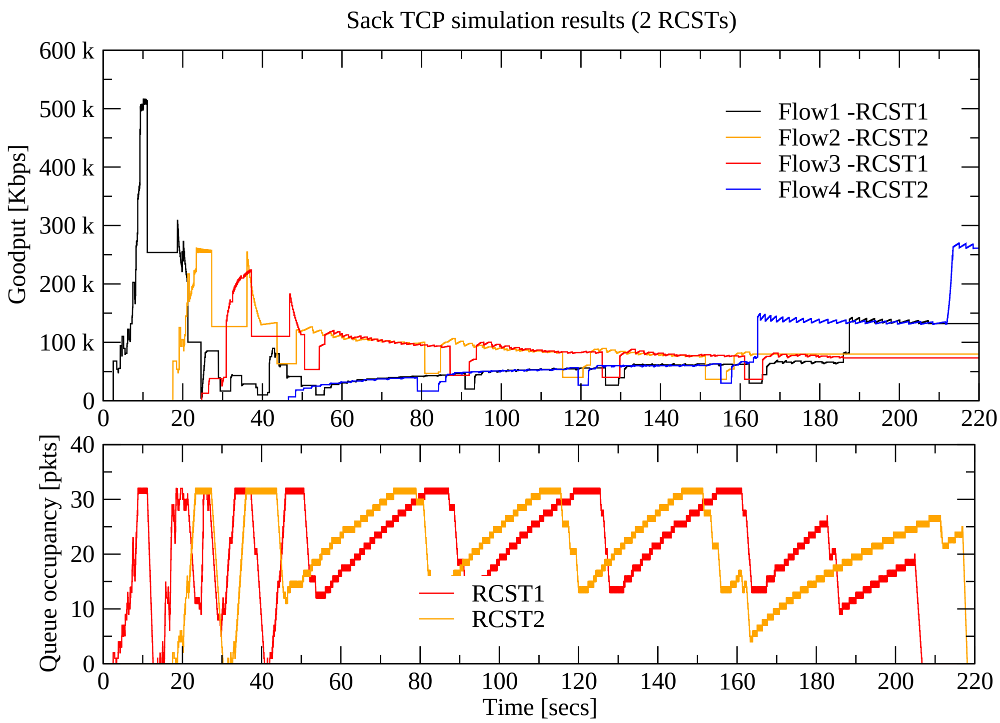


Figure 7.11: Simulation results (four flows) only Sack TCP (Goodput and Queue occupancy)

TCP, low cost of deployment, low signaling overhead (guaranteeing a minimum signaling delay), low implementation complexity (reduced processing time with high scalability), full satellite network exploitation, minimum user response time and fair resource allocation among competing TCP flows. These goals have been quantitatively evaluated by means of simulation. This has been performed using the NS-2 tool in which an implementation of the TCL code has been added for simulating the XPLIT-RCS architecture.

In this context, the XPLIT-RCS architecture can be considered a *complete knowledge* design, as it is possible to control the dynamics of the system, while achieving high system efficiency, enhancing QoS guarantees and providing acceptable fairness level. Moreover, it has been identified that both parameters: *the buffer occupancy and the service rate (β_i and μ_i)* are the only required cross-layer parameters to provide optimized congestion control functions.

As it has been demonstrated in the performance evaluation section, using XPLIT-TCP the transfer times for short-lived connections are reduced, reaching 12.66 % of improvement. In addition, the TCP responsiveness is enhanced when several flows are working together. Using XPLIT-TCP its is possible to reduce the convergence time for several flows compared to Sack TCP in which the convergence time is long slow. Finally, the

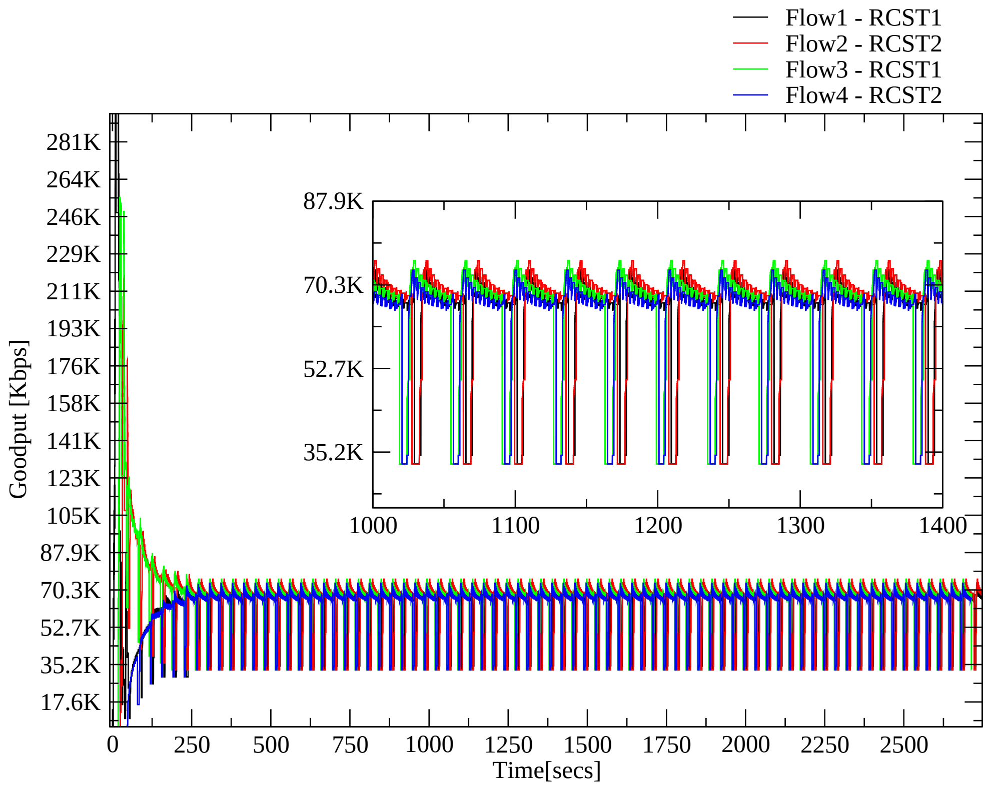


Figure 7.12: Long term simulation results (four flows) only Sack TCP (Goodput)

XPLIT-TCP variant is able to fairly share the DAMA bandwidth allocation with none congestion losses, while having a constant bounded delay per each DAMA queue.

CHAPTER 8

Unified architecture: QoS provisioning over DVB-S2/RCS BSM satellite systems

Nowadays, satellite systems are key elements that enable the provisioning of Internet services over mobile users. Moreover, due to its wide footprint such systems enable connecting distant places where copper/optical infrastructures are not existed, being able to reach a larger proportion of users. Currently, using satellite systems it is possible to reach around 50 % of the world population, however, the target has been estimated to reach up to 90 % by 2020 [ETC$^+$11].

In this way, the main challenge presented in this thesis, which has been focused on the provisioning of QoS guarantees over DVB-S2/RCS satellite systems, is still holds as an important issue that future satellite system designs must consider.

In this chapter, a unified architecture to provide QoS guarantees based on cross-layer design is presented. It takes into account the algorithms and mechanisms proposed by the QoSatArt and XPLIT architectures in order to enhance the interactions among layers forming the ETSI-BSM-QoS protocol stack, while looking for the optimization the satellite system resources.

The main objective of the unified architecture is to provide a reference design in which both architectures (QoSatArt and XPLIT) work jointly to enhance the provisioning of QoS guarantees over satellite systems.

The unified design considers as a central element, the *XL-Manager* module (proposed in the QoSatArt architecture Section 3.3), in order to manage and orchestrate the cross-layer interactions among layers. Additionally, the architecture adopts the cross-layer interactions described in the previous chapters. Firstly, it considers the interaction between the physical layer and the network layer to guarantee the QoS requirements established in the SLA while considering the physical layer adaptability due to the presence of rain events. Secondly, it considers the cross-layer interaction between the network layer and the transport layer, to enhance data transmission based on the TCP protocol.

The final contribution of the presented thesis is focused on specifying a complete architecture design developed for the forward and the return channels, based on the DVB-S2/RCS standard and the ETSI-BSM-QoS protocol stack.

8.1 Scenario for the unified architecture

The scenario for developing the unified architecture is shown on Fig. 8.1. It considers two satellites working in the Ka frequency band (30/20 GHz), in a transparent multi-beam architecture. Here, both GEO satellite systems are used separately for providing interactive services using the DVB-S2 forward channel and the DVB-RCS return channel. In the proposed scenario several remote sources, with different QoS flows, send interactive data to a destination allocated at the Internet side.

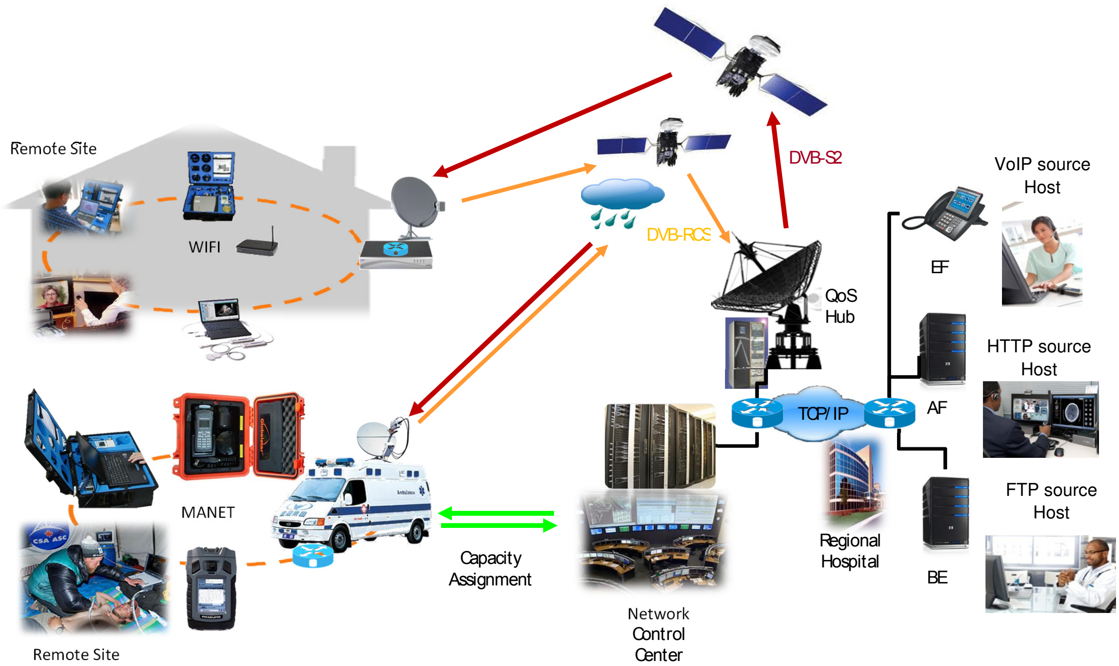


Figure 8.1: Scenario for the Unified architecture

In particular, two emergency remote users (a residential client and a mobile vehicle) require accessing critical applications and data allocated at the regional hospital to provide the first medical aid during an emergency situation (i.e. earthquake, tsunami, etc.), where the satellite system is the only technology that remains available.

In the terrestrial segment, each Return Channel Satellite Terminal (RCST) works together with a PEP which is responsible for managing TCP connections. Each data transmission preformed by the RCST is controlled by the Network Control Center (NCC) to provide interactive applications among users. The NCC allows the support of a Bandwidth-on-demand (BoD) mechanism based on the DAMA protocol, to share the radio spectrum among remote users. The DAMA mechanism provides the capacity assignments requested by each RCST to guarantee the bandwidth availability for the required applications.

On the other hand, the Hub (allocated at the infrastructure site) is defined as the central element for managing the DVB-S2 forward channel connections. On Fig. 8.1, three sources, with different QoS levels, send data to several destination RCSTs by means of the broadcast DVB-S2 GEO satellite channel.

For the provisioning of QoS guarantees, both elements, the Hub and the RCST adopt the ETSI-BSM-QoS architecture based on the DiffServ framework. One of the main goals of having a unified architecture is to guarantee different QoS levels for IP traffic over the DVB-S2/RCS channel, considering the fact that the available bandwidth present in the satellite system is constantly changing due to atmospheric impairments such as rain events. Moreover, it is necessary to consider the effects of adopting transport protocols such as the TCP protocol.

In the proposed unified architecture, the main functional blocks are developed at the terrestrial segment. This is done in order to allow satellite operators to easily adopt the proposed architecture with low deployment cost. In Addition, the proposed architecture allows the satellite operator to manage the functional parameters to establish priority levels and traffic rates according to the defined Service Level Agreements (SLAs).

8.2 Design specifications for the unified architecture

In order to define the essential requirements for the unified architecture, it is important to consider the advantages that both architectures (QoSatArt and XPLIT) are able to provide separately. Therefore, the design goals that each architecture is able to reached are summarized on Table 8.1.

Table 8.1: QoSatArt and XPLIT architecture summary

Design goals	QoSatArt	XPLIT	Unified architecture
a) SI-centric approach	yes	yes	yes
b) QoS provisioning	yes	yes	yes
c) Adaptive physical layer support	yes	no	yes
d) Reduced deployment costs	yes	yes	yes
e) Using the current infrastructure	yes	yes	yes
f) Low signaling overhead	yes	yes	yes
g) Low implementation complexity	yes	yes	yes
h) User-response time	low	low	low
i) Satellite network exploitation	full	full	full
j) Using any TCP variant	no	yes	yes
k) Fairness	yes	yes	yes
l) Friendliness	yes	no	yes

As it is shown, the right column of Table 8.1 summarize the design specifications that the unified architecture must be able to provide. These are described as follows:

a) **Architecture design focussed on SI layers.** The unified architecture considers a design focused on the SI layers (called *SI-centric approach*) following guidelines defined in the ETSI-BSM framework. Here, the *XL-Manager* module is defined as a central element, in order to manage and orchestrate the cross-layer interactions among layers. Using this approach, it is possible to empower the management and control functions performed at the upper layers [M M08], while isolating the SD layers to include different physical layer supports.

b) **QoS provisioning.** For the provisioning of QoS guarantees the design of the unified architecture is based on the ETSI-BSM-QoS [10205] standards including several functional blocks to manage the QoS provisioning. In particular, the DiffServ framework is used to guarantee different QoS levels for IP traffic over the DVB-S2/RCS channels.

c) **Adaptive physical layer support.** The design of the unified architecture consider that the satellite system is affected by atmospheric conditions such as rain events, which reduce the bandwidth availability. Therefore, the adoption of algorithms and cross-layer techniques to guarantee the QoS requirements established in the SLA are specified as mandatory.

d) **Reduced deployment costs.** The unified architecture is designed considering the adoption of algorithms and mechanism working at the terrestrial segment (the Hub and the RCSTs). This is done in order to allow satellite operators to easily adopt the proposed architecture with low deployment costs, while exploiting the DVB-S2/RCS network currently assets.

e) **Exploitation of the current operator infrastructure.** Given that in current infrastructures, most of the RCST terminals work jointly with a PEP element (also called CPEs). The unified architecture is defined considering a PEP element allocate at the Hub, as a mandatory element required to manage DVB-S2 forward connections. Nevertheless, the PEP element allocated at each RCST, is defined to be an optional element. As a result, the unified architecture is designed for single-users or SoHo (Small office/Home office) networks, as it is not necessary to introduce new elements in the end-user subscriber equipment.

f) **Low signaling overhead and minimum signaling delay.** One of the most important issues when working with GEO satellite systems is to prevent signaling traffic passing across the satellite link, as this path is considered the bottleneck link, characterized for being a scarce resource. As a result, the unified architecture is designed considering the exchange of signaling traffic within the LAN network that connects the PEP element either with the HUB or with the RCST, on the terrestrial segment. In addition, to reduce signaling overhead the signaling flows are designed considering a set of aggregated flows based on the DiffServ framework.

g) **Low implementation complexity and low processing.** The unified architecture is designed avoiding either per-TCP-flow or per-packet processing, while using algorithms and mechanism having low computation requirements. It is based on the DiffServ architecture which is able to manage aggregated flows.

h) **Minimum user-response time.** To comply with the Service Level Agreements (SLAs) it is important to reduce the user-response time. Therefore, the unified architecture is designed considering the XPLIT-TCP variant [AMDMn^{+}12] as the transport protocol used to send data through the forward and the return channel. As it was shown by simulation in Section 7.3.3, this XPLIT-TCP has demonstrated to provide reduced user-response times.

i) **Full satellite network exploitation.** The unified architecture is designed focused on maximizing the TCP throughput and the fairness parameters considering the intrinsic characteristics present in GEO satellite systems such as losses, bandwidth variations and long delays. This is done, by adopting the XPLIT-TCP protocol that uses two control loops to achieve full exploitation of the satellite link.

j) **The source and destination hosts are allowed to use standard TCP.** The XPLIT-TCP design is based on the standard TCP specifications, as it implements the standard TCP headers, the TCP standard flow control (based on a sliding window) and the TCP standard error control (based on retransmissions and time-outs). As a result, source and destination hosts are allowed to use either standard TCP variants or XPLIT-TCP.

k) **Fairness and Friendliness among competing TCP flows.** The unified architecture is designed considering an algorithm to share the available resources in a fairly way. This algorithm is able to apply the required policy to share the resources available among several Diffserv classes. This algorithm works jointly with the IP scheduler to control the Diffserv queues, providing them with the suitable *per DiffServ class bandwidth* allocation.

8.3 Unified architecture design

Following the design requirements previously defined, Figure 8.2 shows *the unified architecture design developed for the DVB-S2 forward channel*.

As it is observed, inside the Hub the mechanism called the *XL-Manager*, (which has been proposed in QoSatArt architecture), is considered as central element. This element is able to manage and orchestrate the cross-layer interactions among layers since it is set at the SI upper layers. This modular reference architecture allows the enforcement of control functions performed by the SI layers which can be either modified or updated regardless of the SD layer technology.

In particular, the cross-layer mechanism defined between the satellite physical layer and the SI layer to guarantee the QoS requirements established in the SLA is depicted using red dotted lines on Figure 8.2. One of the main consideration, it is that the satellite system

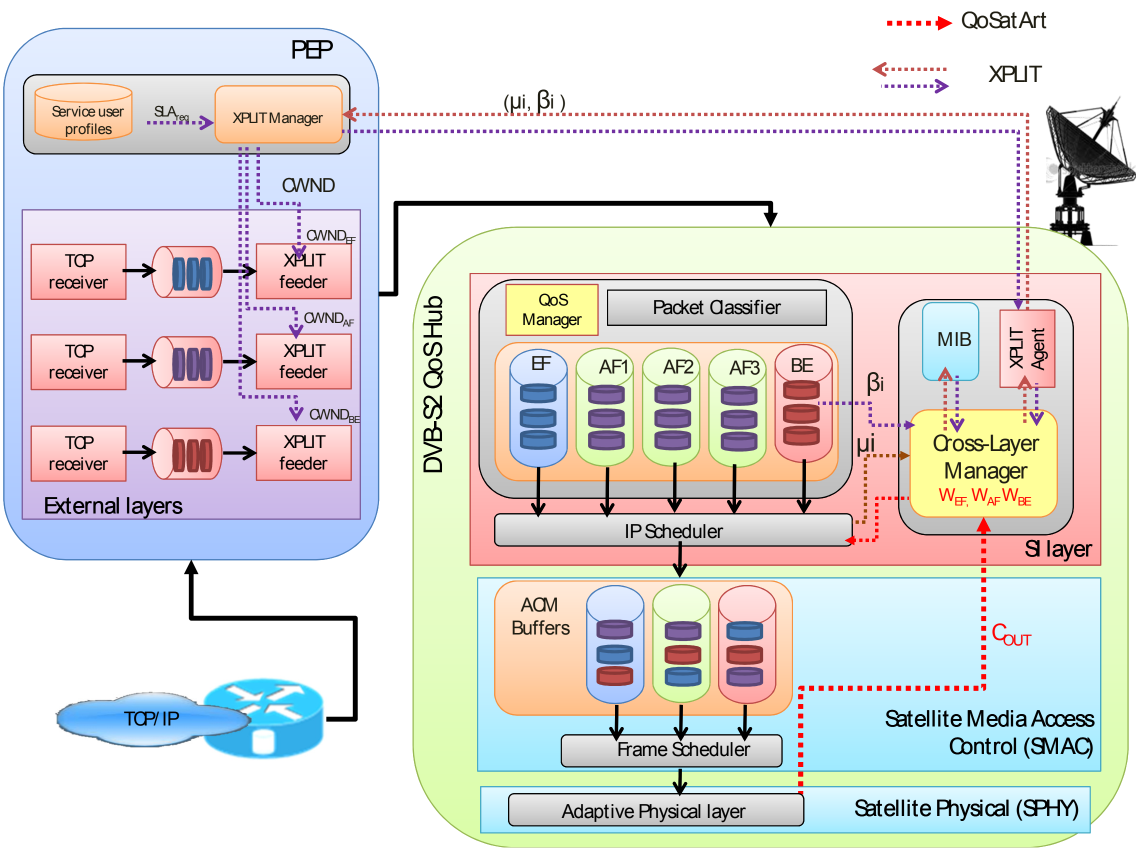


Figure 8.2: The unified architecture design for the DVB-S2 forward channel

is affected by rain events and propagation impairments which can reduce the bandwidth availability.

In this way, the *XL-Manager* prioritizes the resources for high priority traffic classes, by considering different weight values which are calculated and sent to the IP scheduler. As it is observed on Figure 8.2, the *XL-Manager* module is decoupled from the IP scheduler, this means that the scheduler complexity is not increased and both modules can work independently based on their own settings.

On the other hand, the unified architecture considers a second cross-layer mechanism, which is also coordinated by the *XL-Manager*. This cross-layer mechanism is set to share information between the RCST and the PEP in order to improve TCP transmission over the DVB-S2 forward satellite channel. The signaling interactions are defined between the SI layer (at the Hub) and the external/transport layer (at the PEP). This interaction is depicted using purple dotted lines on Fig. 8.2. The physical connection between the RCST and the PEP is performed by means of a Local Area Network (LAN).

In particular, the transport protocol used to send data through the forward and the return channel, is the XPLIT-TCP variant [AMDMn+12] in order to enhance data transmission based on the TCP protocol. Here, the PEP uses *TCP feeders* to allow interoperability between standard TCP versions and the XPLIT-TCP protocol. In this way, the end-users are directly connected to the PEP to send data through the Hub, by means of the broadcast DVB-S2 GEO satellite channel.

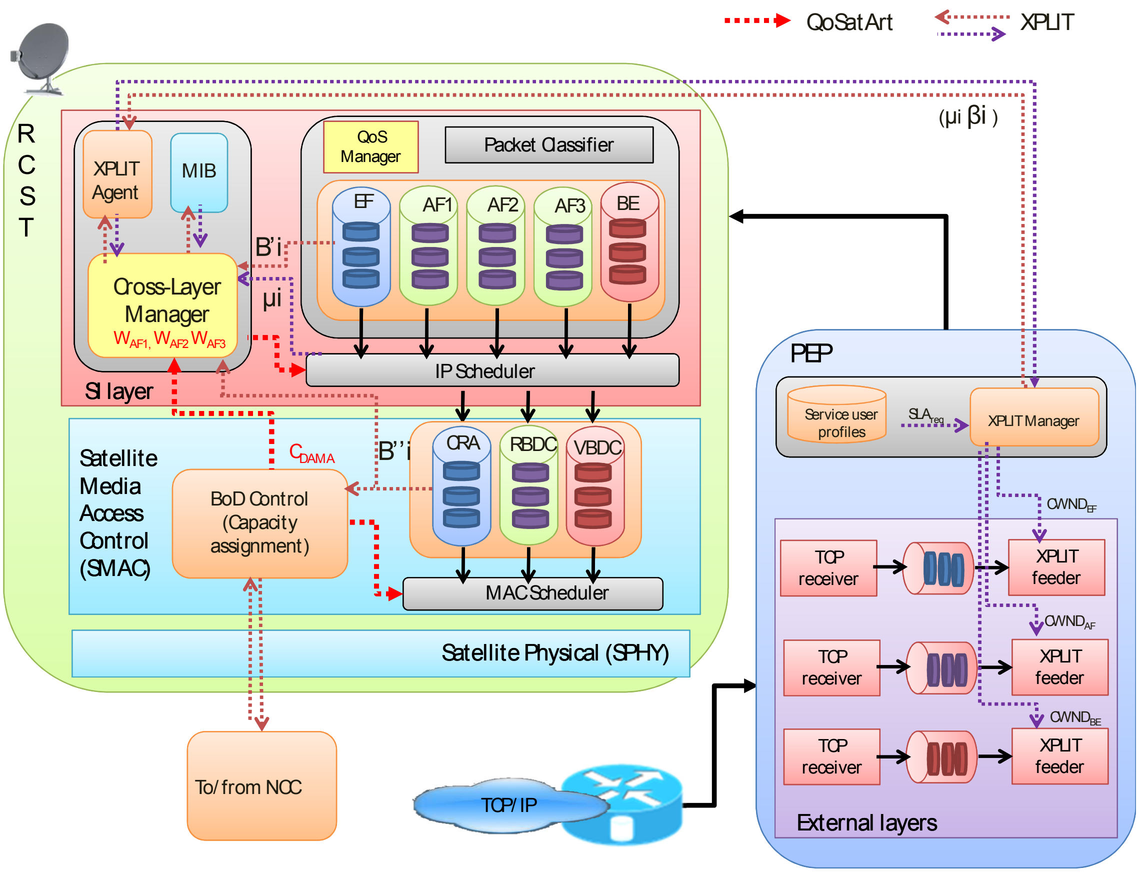


Figure 8.3: The unified architecture design for the DVB-RCS return channel

Inside the PEP (see Fig. 8.2), the XPLIT Manager is responsible for adjusting the *cwnd* values for each TCP flow. The unified architecture also includes the design of a signaling mechanism aiming to control the cross-layer parameters between the PEPs and either the Hub or the RCSTs.

This design requires two main cross-layer parameters set up at the Hub: the buffer occupancy and the service rate (β_i and μ_i). The buffer occupancy is measured for each QoS queue while the service rate is assessed at the IP scheduler, both of them set at the the SI layer (see Fig. 8.2). Both parameters (β_i and μ_i) are forwarded to the PEP element using a cross-layer mechanism. Here, the transport layer is responsible for receiving the cross-layer parameters which are used to modify and adjust the *cwnd* values accordingly for each TCP flow. The cross-layer design and the exchange of the parameters β_i and μ_i are also shown on Fig. 8.2.

Similarly, *the unified architecture design developed for the DVB-RCS return channel* is depicted on Fig. 8.3. As it is shown on Fig. 8.3, incoming flows are received by the PEP to be forwarded through the RCST to the destination host by means of the DVB-RCS return satellite channel. As it is observed, the architecture developed for the DVB-RCS return satellite channel has been provided with the same modules previously described. However, it includes the modules associated with the NCC to allow the support of a Bandwidth-on-demand (BoD) mechanism based on the DAMA protocol. The DAMA mechanism provides the capacity assignments requested by each RCST to guarantee the bandwidth availability for the required applications.

Finally, the *E2E unified architecture design developed for both the DVB-S2/RCS forward and return channel* is depicted on Fig. 8.4.

8.4 Conclusions

In this chapter the unified architecture design has been proposed. It considers the advantages that the XPLIT together with QoSatArt architectures can provide to enhance TCP transmissions while guaranteeing the predefined QoS specifications. The unified architecture considers the cross-layer design developed for XPLIT to share information between the RCST and the PEP. This improves the TCP transmission over the forward and the return satellite channel. Moreover, it considers the cross-layer design developed for the QoSatArt to enhance QoS provisioning considering the bandwidth availability present at the satellite physical layer.

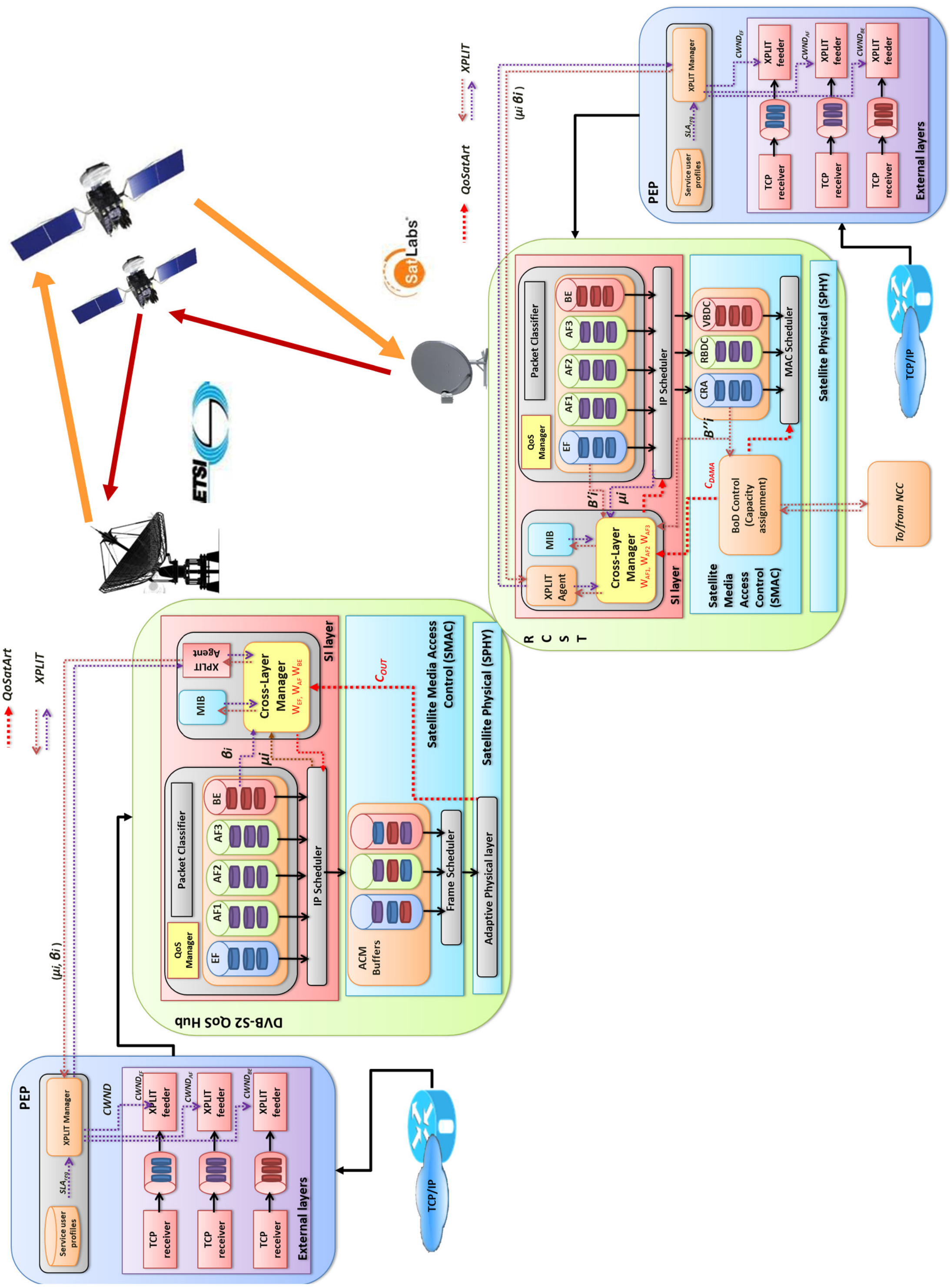


Figure 8.4: The unified E2E architecture for QoS support over BSM satellite systems

CHAPTER 9

Overall conclusions

The presented thesis has been focused on researching about the interactions occurring among layers in order to design and evaluate a complete architecture for providing QoS guarantees over DVB-S2/RCS satellite systems.

In this thesis several contributions based on cross-layer design to enhance the provisioning of QoS guarantees over Broadband Satellite systems have been presented. The proposed design considers the drawbacks posed by GEO satellite systems while implementing cross-layer techniques to enhance QoS provisioning.

In particular, the contributions have been designed following the specifications defined in the ETSI-BSM-QoS framework [10205]. In addition, the advantages that the *SI-centric* design can provide are analyzed. As it can enhance the management functions performed at the satellite upper layers [M M08], while isolating the satellite lower layers to include different physical layer supports.

Hence, it can be concluded that the main goal of this PhD thesis, focused on improving BSM satellite systems with the support of QoS guarantees by means of cross-layer techniques, has been successfully reached. In this way, the methodology defined to reach this goal is probably the most important result that has been obtained during the development of this PhD thesis. Moreover, the architecture design and the simulation results to enhance the QoS provisioning over satellite systems have also been a valuable outcome.

The initial analysis has been focused on studying the cross-layer interactions among layers forming the ETSI-BSM protocol stack. It includes the state-of-the-art study, that analyzes the interactions among layers. Secondly, to identify the possible ways to enhance QoS provisioning, two main architectures have been proposed: QoSatArt and XPLIT. These designs take into account the drawbacks posed by GEO satellite systems while looking for the optimization of the satellite system resources. Finally, the implementation and evaluation of the proposed architectures using the NS-2 simulation tool have been performed. This has been done to demonstrate the advantages that the proposed solutions can provide to enhance satellite communications with QoS support.

Figure 9.1 shows the main contributions developed in this PhD Thesis for the XPLIT and QoSatArt architectures. Here, the adoption of cross-layer techniques is also depicted in order to show schematically the main characteristics present in the design of the proposed architectures.

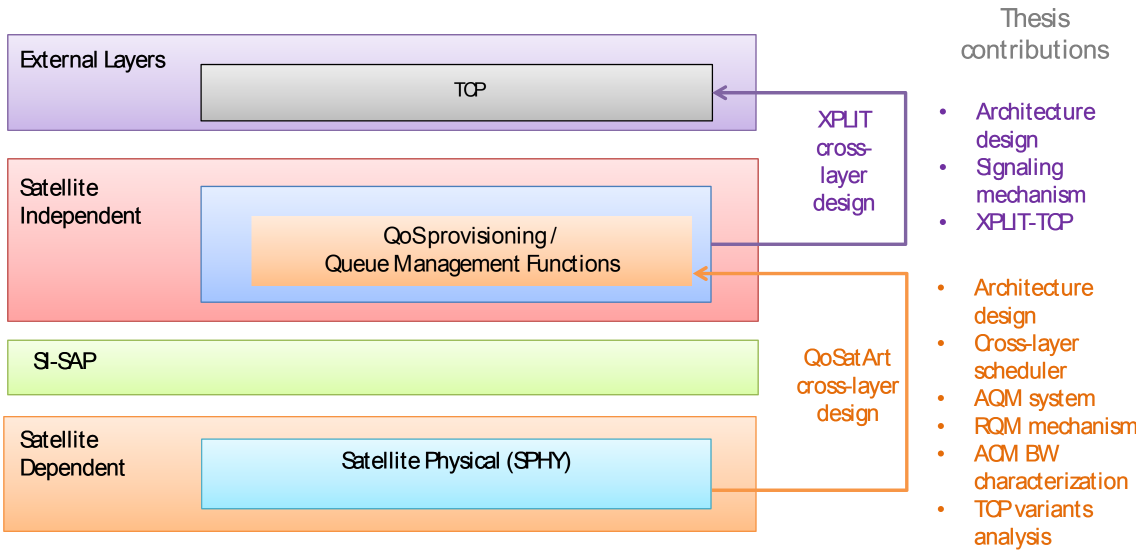


Figure 9.1: Thesis contributions

The first contribution, analyzes the interaction between the SD lower layers and the SI network layer. Here, the QoSatArt architecture has been designed to provide QoS guarantees considering the physical layer adaptability. It includes the performance analysis of several standard TCP variants aiming to find the most suitable TCP variant that enhances TCP transmission over the proposed architecture.

The second contribution, studies the relationship between the SI network layer and the transport layer to enforce QoS provisioning across the upper layers of the ETSI-BSM protocol stack. Here, the XPLIT architecture has been proposed to enhance TCP transmission and provide them with QoS guarantees. The XPLIT architecture is developed for the forward and the return channels, together with a modified TCP variant called XPLIT-TCP used to enhance TCP transmissions.

9.1 Conclusions for QoSatArt architecture

As it has been mentioned before, the QoSatArt architecture has been proposed to provide QoS guarantees for IP traffic over the DVB-S2 satellite channel. It includes a cross-layer optimization between the physical layer and the network layer to guarantee the QoS requirements established in the SLA. Here, the main design consideration is the adoption of a centralize mechanism called the *XL-Manager*, which is defined to manage and orchestrate the cross-layer interactions among layers as it is set up at the SI upper layers. In addition, the design considers that the satellite system is affected by atmospheric events, such as rain, which reduce the bandwidth availability.

Moreover, as it is observed on Fig. 9.1, the architecture includes the design of *the Active Queue Management (AQM)* system to minimize the delay values experienced at the Application layer for delay sensitive traffic. Here, a complete QoS design inside the *gateway* has been detailed, in which *the RQM mechanism* is proposed to enhance the goodput for the EF and AF traffic classes.

In addition, *a cross-layer IP scheduler* has been proposed to guarantee the high priority traffic classes regardless of the channel condition affected by rain events. The available bandwidth is determined considering the ACM adaptation behaviour reported by each RCST terminal. Here, *a characterization of the link bandwidth variations* has been performed using a sinusoidal approximation to determine the intensity of the rain event affecting the bandwidth availability.

The algorithm for calculating the weighted values to provide QoS guarantees proposes a modification based on the PDS model which includes an exponential function to associate the QoS requirements for each traffic class with the satellite link capacity. The QoSatArt architecture proposes a complete design defined at the terrestrial segment (either at the gateway or at the Customer Premises Equipment (CPE)) in order to provide low complexity on its implementation and seamlessly inter-operate with terrestrial IP networks.

The key results obtained when evaluating the QoSatArt architecture show that it is possible to keep control of the satellite system load while guaranteeing QoS levels for the high priority traffic classes even though bandwidth variations due to rain events are experienced. The simulation results also demonstrate that with the adoption of the proposed RQM mechanism, the user's QoE is improved while keeping bounded the delay and jitter values for high priority traffic classes. In particular an AF goodput enhancement of 33% (on average) is reached. Moreover, with the evaluation of the cross-layer IP scheduler, the high priority traffic class is always guaranteed regardless of the channel conditions affected by rain events.

Here, the most important aspect in the QoSatArt it is that the proposed architecture has been designed to guarantee the QoS requirements for specific traffic flows using a single parameter: the bandwidth availability. This parameter is set at the physical layer (considering ACM adaptation) and sent to the IP scheduler taking advantage of the proposed cross-layer optimization.

Finally as a final contribution within the context of QoSatArt, *an analysis of several TCP variants* has been performed (see Figure 9.1). Here, CUBIC TCP, Hybla TCP and Sack TCP are the selected TCP variants to carry out the performance evaluation.

The TCP analysis has been performed using metrics such as goodput, queue occupancy, delay and jitter values in order to analyze the friendliness and fairness behavior of the interaction of the selected TCP variants.

The simulation results have shown that Hybla TCP provides an aggressive behavior which adversely affects the overall satellite performance. This is mainly caused because of the implementation of the RQM mechanism that enables the allocation of a considerable amount of out-of-profile packets in the BE queue. On the other hand, Sack TCP shows good fairness and friendliness level, as it is able to fairly share the bandwidth when constant link capacity is available. CUBIC TCP is able to provide a stable goodput level for the AF traffic class while keeping the queue occupancy at lower levels when a DVB-S2 satellite system experiences bandwidth reductions.

Therefore, it can be concluded that when the proposed QoSatArt works together with the adoption of the CUBIC TCP variant, it is possible to obtain an enhanced TCP performance when the satellite link experiences severe bandwidth reductions due to the presence of rain events while guaranteeing the required QoS specifications defined on the whole E2E path.

9.2 Conclusions for XPLIT architecture

The second contribution of this thesis has been focused on developing an architecture, based on cross-layer design, to enhance the performance of the TCP protocol over the satellite segment. As it is seen on Fig. 9.1, the XPLIT architecture has been proposed to enhance TCP transmission with QoS for DVB-S2/RCS satellite systems.

This architecture considers the challenges posed by GEO satellite systems (i.e. delay, losses and bandwidth variations) while proposing several mechanisms based on cross-layer design to enhance the performance of the TCP protocol.

The architecture proposes the use of Performance Enhanced Proxies (PEPs) which breaks the End-to-End semantic of TCP connections. However, XPLIT architecture considers a cross-layer design between the network layer and the transport layer to enhance data transmission. It includes the design of a signaling mechanism aiming to control the cross-layer parameters between the PEPs and either the Hub or the RCSTs. The design is developed for the forward and the return channels, which are based on the DVB-S2/RCS standards. It considers the Bandwidth on Demand (BoD) mechanism to dynamically assign the system resources. The XPLIT architecture has been designed to provide QoS guarantees over the DVB-S2/RCS based on the ETSI-BSM QoS architecture for satellite systems.

The proposed architecture also adopts the XPLIT-TCP variant to provide enhanced results when working with the DAMA capacity assignments. Moreover, it has been identified that both parameters: *the buffer occupancy and the service rate (β_i and μ_i)* are the only required cross-layer parameters to provide optimized congestion control functions.

The XPLIT-TCP variant is able to fairly share the DAMA bandwidth allocation allowing reduced transfer times for short and long-lived connections. In addition, the TCP responsiveness for the return link is enhanced when several flows are working together. Using XPLIT-TCP its is possible to reduce the convergence time for several flows compared to Sack TCP. Finally, the XPLIT-TCP variant is able to fairly share the DAMA bandwidth allocation with none congestion losses, while having a constant bounded delay per each DAMA queue.

In this context, the XPLIT architecture can be seen as a design able to have the *complete knowledge* of the system characteristics, as it is possible to control the dynamics of the system, while achieving high system efficiency, enhancing QoS guarantees and providing acceptable fairness levels.

9.3 Final conclusions

As final contribution, an unified architecture to provide QoS guarantees based on cross-layer design has been proposed. It takes into account the algorithms and mechanisms proposed by the QoSatArt and XPLIT architectures in order to enhance the interactions among layers forming the ETSI-BSM-QoS protocol stack, while looking for the optimization the satellite system resources.

The unified design considers as a central element, the *XL-Manager* module in order to manage and orchestrate the cross-layer interactions among layers. Additionally, the architecture adopts the cross-layer interactions in order to comply with the following design requirements:

The unified framework is designed considering a SI-centric approach, in order to provide QoS guarantees, by taking into account the Adaptive physical layer characteristics present on GEO satellite systems. The architecture is designed to Reduce the deployment costs, which can be implemented by satellite operators in their current infrastructures. Moreover, the design allows Low signaling overhead and Low implementation complexity. Here, the minimum User-response time is guaranteed while achieving full satellite network exploitation. Finally, the design allows the use of any TCP variant in order to guaranteeing Fairness and Friendliness among TCP flows.

These goals have been separately evaluated by means of simulation for QoSatArt and XPLIT architectures.

9.4 Future prospects

As final note, it is worth mentioning some futures research areas directly related to the work presented in the presented thesis. Firstly, the Unified architecture to provide QoS guarantees could be simulated and evaluated using either a simulating or an emulating platform. Moreover, the implementation of the proposed Unified architecture working over a real DVB-RCS satellite receptor equipment will be a valuable future work. The support of QoS provisioning over heterogeneous networks considering standards such as DVB-RCS + M and DVB-NG including the adoption of cross-layer techniques remains as important topic for the future.

Bibliography

[10205] ETSI TS 102 462. ETSI 102 462: Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM); Services and Architectures: QoS Functional Architecture , 2005.

[10207] ETSI TS 102 464. ETSI 102 464 Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM); Interworking with DiffServ QoS , 2007.

[18505] ETSI TS 185 001. Telecommunication and Internet converged Services and Protocols for Advanced Networking (TISPAN); Next Generation Network (NGN); Quality of Service (QoS) Framework and Requirements. 2005.

[30097] ETSI EN 300 421. ETSI Digital Video Broadcasting (DVB); Second Generation. Framing structure, channel coding and modulation system for 11/12 Ghz Satellite Services, august 1997. ETSI Specification: ETSI EN 300 421.

[3GP99] 3GPP TS 23.107, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Quality of Service (QoS) concept and architecture (Release 7). Technical Report V7.1.0, 3GPP, 1999.

[AM04] Barsaleau Dean A. and Tummala Murali. Testing of diffserv performance over a u.s. navy satellite communication network. *IEEE Military Communications Conference*, 2004.

[AMDMn+12] Juanjo Alins, Jorge Mata-Diaz, Jose L. Muñoz, Elizabeth Rendon-Morales, and Oscar Esparza. Xplit: A cross-layer architecture for tcp services over dvb-s2/etsi qos bsm. *Comput. Netw.*, 56(1):412–434, January 2012.

[AVCV12] M. Angeles Vazquez Castro and F. Vieira. Dvb-s2 full cross-layer design for qos provision. *Communications Magazine, IEEE*, 50(1):128 –135, january 2012.

[AZ12] Seyed Mohamad Alavi and Chi Zhou. Resource allocation scheme for orthogonal frequency division multiple access networks based on cooperative game theory. *International Journal of Communication Systems*, pages n/a–n/a, 2012.

[BAFW03] E. Blanton, M. Allman, K. Fall, and L. Wang. A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP. RFC 3517, Internet Engineering Task Force, April 2003.

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Service. RFC 2475, Internet Engineering Task Force, December 1998.

[BCS94] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC 1633, Internet Engineering Task Force, June 1994.

[BSAD04] S. Bohacek, K. Shah, G.R. Arce, and M. Davis. Signal processing challenges in active queue management. *Signal Processing Magazine, IEEE*, 21(5):69 – 79, sept. 2004.

[BY04] R.A. Berry and E.M. Yeh. Cross-layer wireless resource allocation. *Signal Processing Magazine, IEEE*, 21(5):59 – 68, sept. 2004.

[CABG05] Stéphane Combes, Olivier Alphand, Pascal Berthou, and Thierry Gayraud. Satellite and next generation networks: proposal for qos architectures. *Space Comms.*, 20:101–119, August 2005.

[CFL+09] C. Caini, R. Firrincieli, D. Lacamera, T. de Cola, M. Marchese, C. Marcondes, M. Y. Sanadidi, and M. Gerla. Analysis of tcp live experiments on a real geo satellite testbed. *Perform. Eval.*, 66(6):287–300, 2009.

[cFSKS02] Wu chang Feng, K.G. Shin, D.D. Kandlur, and D. Saha. The blue active queue management algorithms. *Networking, IEEE/ACM Transactions on*, 10(4):513 – 528, aug 2002.

[CG07] M.A.V. Castro and G.S. Granados. Cross-layer packet scheduler design of a multibeam broadband satellite system with adaptive coding and modulation. *Wireless Communications, IEEE Transactions on*, 6(1):248 –258, jan. 2007.

[CGR08] S. Cioni, R. Gaudenzi, and R. Rinaldo. Channel estimation and physical layer adaptation techniques for satellite networks exploiting adaptive coding and modulation. *International Journal of Satellite Communications and Networking*, 26(2):157–188, 2008.

[CKP09] Wei Koong Chai, Merkourios Karaliopoulos, and George Pavlou. Providing proportional tcp performance by fixed-point approximations over bandwidth on demand satellite networks. *Trans. Wireless. Comm.*, 8(7):3554–3565, July 2009.

[CLF+06] C. Caini, N.C. Liberato, R. Firrincieli, , and G. Giambene. TCP Hybla Performance in GEO Satellite Networks: Simulations and Testbed. In *Proc. International Workshop on Satellite and Space Communications*, pages 41–45, 14–15 Sept. 2006.

[DCB+02] B. Davie, A. Charny, J.C.R. Bennet, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis. An Expedited Forwarding PHB (Per-Hop Behavior). RFC 3246, Internet Engineering Task Force, March 2002.

[DKG+01] Arjan Durresi, Sastri Kota, Mukul Goyal, Raj Jain, and Venkata Bharani. Achieving qos for tcp traffic in satellite networks with differentiated services, 2001.

[DKZsM05] Ita Dukkipati, Masayoshi Kobayashi, Rui Zhang-shen, and Nick Mckeown. *Processor sharing flows in the internet.* "", 2005.

[DP06] Wei D. and Cao P. A linux tcp implementation for ns2. *available at: www.cs.caltech.edu / weixl/technical/ns2linux/index.html*, 2006.

[DR99] C. Dovrolis and P. Ramanathan. A case for relative differentiated services and the proportional differentiation model. *Network, IEEE*, 13(5):26 –34, sep/oct 1999.

[DSC+05] A. Durresi, M. Sridharan, S. Chellappan, R. Jain, H. Ozbai, and L. Barolli. Control theory optimization of mecn in satellite networks. In *Distributed Computing Systems Workshops, 2005. 25th IEEE International Conference on*, pages 770 – 776, june 2005.

[DSLG01] Arjan Durresi, Mukundan Sridharan, Chunlei Liu, and Mukul Goyal. Congestion control using multilevel explicit congestion notification in satellite networks. 1(1):483–488, August 2001.

[DSLJ01] Arjan Durresi, Mukundan Sridharan, Chunlei Liu, and Raj Jain. Improved explicit congestion notification for satellite networks. In *Proceedings of SPIE International Conference on Quality of Service over the Net Generation Data Networks*, volume 4524, pages 293–303, August 2001.

[DSR02] Constantinos Dovrolis, Dimitrios Stiliadis, and Parameswaran Ramanathan. Proportional differentiated services: Delay differentiation and packet scheduling. In *IEEE/ACM Transactions on Networking*, pages 109–120, 2002.

[ECB08] A-L. Beylot E. Chaput and C. Baudoin. Packet scheduling over dvb-s2 through gse encapsulation. In *IEEE Global Telecommunications Conference*, pages 1–5, 30 2008-dec. 4 2008.

[ECC02] Ec cost 255, radiowave propagation modelling for new satellite communication services at ku-band and above. *ESA Publications Division, EC COST 255 Final Report*, (SP-1252), March 2002.

[esa07] IP-friendly cross-layer optimization of adaptive satellite systems, 2007. European Spatial Agency (ESA) Contract 19237/05/NL/AD.

[ETC+11] B.G. Evans, P.T. Thompson, G.E. Corazza, A. Vanelli-Coralli, and E.A. Candreva. 1945-2010: 65 years of satellite history from early visions to latest missions. *Proceedings of the IEEE*, 99(11):1840 –1857, nov. 2011.

[ETS03] ETSI. Satellite earth stations and systems (ses); broadband satellite multimedia (bsm); ip interworking over satellite; performance, availability and quality of service. *ETSI TS 102 157 standard*, 2003.

[ets04] Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2), 2004. ETSI Specification: EN 302 307.

[ets05a] Digital Video Broadcasting (DVB); Interaction Channel for Satellite Distribution Systems; (DVB-RCS), 2005. ETSI Specification: EN 301 790.

[ETS05b] ETSI. Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM); Common Air interface specification; Satellite Independent Service Access Point SI-SAP, 2005.

[FL06] L. Wu F. Peng and V. C. M. Leung. Cross-layer enhancement of TCP split-connections over satellites links. *International Journal of Satellite Communications and Networking*, 24:405–418, 2006.

[For03] ADSL Forum. Technical Report: TR-059: DSL Evolution - Architecture Requirements for the Support of QoS-Enabled IP Services. 2003.

[FVG06] M. A. Vázquez Castro F. Vieira and G. Seco Granados. A tunable-fairness cross-layer scheduler for dvb-s2. *International Journal of Satellite Communications and Networking*, 1(1):255 –270, jan. 2006.

[GH09] G. Giambene and S. Hadzic. A Cross-Layer PEP for DVB-RCS Networks. In *Lecture Notes of the Institute for Computer Sciences. Personal Satellite Services International Conference*, volume 15, pages 12–19, 2009.

[Gia07] Gionanni Giambene. In *Resource Management in Satellite Networks, Optimization and Cross-Layer Design*, volume 1, pages 1–330. Springer, 2007.

[GPS06] A. Gotta, F. Potorti, and R. Secchi. An analysis of tcp startup over an experimental dvb-rcs platform. In *Satellite and Space Communications, 2006 International Workshop on*, pages 176 –180, sept. 2006.

[GWE+09] Mathieu Gineste, Nicolas Wambeke, Ernesto Exposito, Christophe Chassot, and Laurent Dairaine. A cross-layer approach to enhance qos for multimedia applications over satellite. *Wirel. Pers. Commun.*, 50(3):305–328, August 2009.

[HBWW99] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. RFC 2597, Internet Engineering Task Force, June 1999.

[HPW02] D. Harrington, R. Presuhn, and B. Wijnen. An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. RFC 3411, Internet Engineering Task Force, December 2002.

[HR02] D. Katabi M. Handley and Ch. Rohrs. Congestion control for high bandwidth-delay product networks. In *SIGCOMM*, pages 89–102, 2002.

[HRX05] Sangtae Ha, Injong Rhee, and Lisong Xu. Cubic: A new tcp-friendly high-speed tcp variant, 2005.

[Hus00] G. Huston. Next Steps for the IP QoS Architecture. RFC 2990, Internet Engineering Task Force, November 2000.

[ipe05] Interoperable Performance Enhancement Proxy (I-PEP) specification, 2005. SatLabs Group.

[IT04] ITU-T. Recommendation Y.1291 - An architectural framework for support of quality of service (QoS) in packet networks. 2004.

[ITI02] Y. Ito, S. Tasaka, and Y. Ishibashi. Variably weighted round robin queueing for core ip routers. In *Performance, Computing, and Communications Conference, 2002. 21st IEEE International*, pages 159 –166, 2002.

[JBB92] V. Jacobson, R. Braden, and D. Borman. TCP Extensions for High Performance. RFC 1323, Internet Engineering Task Force, May 1992.

[JNP99] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. RFC 2598, Internet Engineering Task Force, June 1999.

[KAE+07] Maria Kalama, Guray Acar, Barry Evans, Stephane Mourgues, Milena Plannels, and Christophe Donny. Cross-layer design and assessment of tcp congestion control protocols over future broadband satellite systems. *25th AIAA International Communications Satellite Systems Conference*, April 2007.

[KGC07] S. Kota, G. Giambene, and N.L. Candio. Cross-layer approach for an air interface of GEO satellite communication networks. *International Journal of Satellite Communications and Networking*, 25(5):481–499, 2007.

[KK05] V. Kawadia and P.R. Kumar. A cautionary perspective on cross-layer design. *Wireless Communications, IEEE*, 12(1):3 – 11, feb. 2005.

[LC04] Ki-Dong Lee and Kun-Nyeong Chang. A real-time algorithm for timeslot assignment in multirate return channels of interactive satellite multimedia networks. *Selected Areas in Communications, IEEE Journal on*, 22(3):518 – 528, april 2004.

[LM97] T.V. Lakshman and U. Madhow. The performance of tcp/ip for networks with high bandwidth-delay products and random loss. *Networking, IEEE/ACM Transactions on*, 5(3):336 –350, jun 1997.

[LRF09] M. Luglio, C. Roseti, and F.Zampognaro. Perfromance Evaluation of TCP-based application over DVB-RCS DAMA schemes. *International Journal of Satellite Communications and Networking*, 27:163–191, 2009.

[LYZ+10] Yajun Li, Yuhang Yang, Liang Zhou, Anne Wei, and Chengyu Cao. Qos-aware fair packet scheduling in ieee 802.16 wireless mesh networks. *International Journal of Communication Systems*, 23(6-7):901–917, 2010.

[LZG05a] Qingwen Liu, Shengli Zhou, and G.B. Giannakis. Cross-layer scheduling with prescribed qos guarantees in adaptive wireless networks. *Selected Areas in Communications, IEEE Journal on*, 23(5):1056 – 1066, may 2005.

[LZG05b] Qingwen Liu, Shengli Zhou, and G.B. Giannakis. Queuing with adaptive modulation and coding over wireless links: cross-layer analysis and design. *Wireless Communications, IEEE Transactions on*, 4(3):1142 – 1153, may 2005.

[LZM07] M Luglio, F Zampognaro, and F Morell T amd Vieira. Joint dama-tcp protocol optimization through multiple cross layer interactions in dvb rcs scenario. *Satellite and Space Communications, 2007*, 14(4):121–125, 2007.

[M M08] M Marchese and M Mongelli. Protocol structure overview of qos mapping over satellite networks. *IEEE ICC proceedings*, 2008.

[Mar03] M. Marchese. Tcp/ip - based protocols over satellite systems: A telecommunication issue, 2003. chapter of book.

[Mar07] Mario Marchese. In *QoS over Heterogeneous Networks*, volume 1, pages 1–328. John Wiley - Sons Ltd, 2007.

[MM04] M. Rossi M. Marchese and G. Morabito. PETRA: performance enhancing transport architecture for satellite communications. *IEEE Journal on Selected Areas in Communications*, 22(2):320–332, 2004.

[MM06] A Morello and V Mignone. Dvb-s2: The second generation standard for satellite broad-band services, 2006.

[MSMO97] Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi, and Teunis Ott. The macroscopic behavior of the tcp congestion avoidance algorithm. *SIGCOMM Comput. Commun. Rev.*, 27(3):67–82, July 1997.

[OK04] J. Saarto O. Riva and M. Kojo. Performance analysis on HTTP traffic and traffic mixtures with competing tcp and udp flows, 2004. IIP Mixture project report.

[Pa10] D.F. Pinas andas. Telecom i+d05 - aine: An ip network emulator. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 8(2):194 –198, april 2010.

[PAPC07] D.K. Petraki, M.P. Anastasopoulos, A.D. Panagopoulos, and P.G. Cottis. Dynamic resource calculation algorithm in mf-tdma satellite networks. In *Mobile and Wireless Communications Summit, 2007. 16th IST*, pages 1 –5, july 2007.

[PEBS00] Peter Pieda, Jeremy Ethridge, Mandeep Baines, and Farhan Shallwani. A network simulator differentiated services implementation. *Open IP, Nortel Networks*, July 2000.

[PFTK98] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose. Modeling tcp throughput: a simple model and its empirical validation. *SIGCOMM Comput. Commun. Rev.*, 28(4):303–314, October 1998.

[PIA+05] Antonio Pietrabissa, Tiziano Inzerilli, Olivier Alphand, Pascal Berthou, Thierry Gayraud, Michel Mazzella, Eddy Fromentin, and Fabrice Lucas. Validation of a qos architecture for dvb-rcs satellite networks via the satip6 demonstration platform. *Comput. Netw.*, 49:797–815, December 2005.

[Pre02] R. Presuhn. Management Information Base (MIB) for the Simple Network Management Protocol (SNMP). RFC 3418, Internet Engineering Task Force, December 2002.

[RG04] Rita Rinaldo and Riccardo De Gaudenzi. Capacity analysis and system optimization for the forward link of multi-beam satellite broadband systems exploiting adaptive coding and modulation. *International Journal of Satellite Communications and Networking*, 22(3):401–423, 2004.

[RLZ09] C. Roseti, Michele Luglio, and Francesco Zampognaro. Analysis and performance evaluation of a burst-based tcp for satellite dvb rcs links. *IEEE/ACM Transactions on Betworking*, 2009.

[RMAMD+13] Elizabeth Rendon-Morales, Juanjo Alins, Jorge Mata-Diaz, Oscar Esparza, and Jose L. Muñoz. Xplit.rcs: A cross-layer architecture for tcp services over dvb-rcs/etsi qos bsm. *International Journal of Satellite Communications*, Submitted, January 2013.

[RMMDA+09] E. Rendon-Morales, J. Mata-Diaz, J. Alins, J.L. Munoz, and O. Esparza. Cross-layer architecture for tcp splitting in the return channel over satellite networks. In *Wireless Communication Systems, 2009. ISWCS 2009. 6th International Symposium on*, pages 225 –229, sept. 2009.

[RMMDA+12a] Elizabeth Rendon-Morales, Jorge Mata-Díaz, Juanjo Alins, Jose L. Muñoz, and Oscar Esparza. Adaptive ip scheduler design to support qos guarantees over satellite systems. *Journal of Internet Engineering*, pages n/a–n/a, 2012.

[RMMDA+12b] Elizabeth Rendon-Morales, Jorge Mata-Díaz, Juanjo Alins, Jose L. Muñoz, and Oscar Esparza. Cross-layer packet scheduler for qos support over dvb-s2 bsm satellite systems. *To be published in the International Journal of Communication Systems*, pages n/a–n/a, 2012.

[RMMDA+12c] Elizabeth Rendon-Morales, Jorge Mata-Díaz, Juanjo Alins, Jose L. Muñoz, and Oscar Esparza. Performance evaluation of selected transmission control protocol variants over a digital video broadcasting-second generation broadband satellite multimedia system with qos. *International Journal of Communication Systems*, pages n/a–n/a, 2012.

[RMMDA+12d] Elizabeth Rendon-Morales, Jorge Mata-Díaz, Juanjo Alins, Jose L. Muñoz, and Oscar Esparza. Qosatar: a cross-layer architecture for e2e qos provisioning over dvb-s2 broadband satellite systems. *EURASIP Journal on Wireless Communications and Networking*, pages n/a–n/a, 2012.

[RPDRF03] L. S. Ronga, T. Pecorella, E. Del Re, and R. Fantacci. A gateway architecture for ip satellite networks with dynamic resource management and diffserv qos provision. *International Journal of Satellite Communications and Networking*, 21(4-5):351–366, 2003.

[RVCM04] R. Rinaldo, M.A. Vazquez-Castro, and A. Morello. DVB-S2 ACM modes for IP and MPEG unicast applications. *International Journal of Satellite Communications and Networking*, 22:367–399, 2004.

[SA10] Shyam Parekh Mostafa Tofighbakhsh Sasan Adibi, Raj Jain. In *Quality of Service Architectures for Wireless Networks: Performance Metrics and Management*, volume 1, pages 1–715. Information Science Publishing, 2010.

[Saa03] J. Saarto. WWW Traffic Performance in Wireless Environment. Master's thesis, University of Helsinki, Department of Computer Science, No. C-2003-35., 2003.

[Sat08] SatLabs System Recommendations - Quality of Service specifications, 2008. http://www.satlabs.org.

[SBB06] Data System Standards, Blue Book, and Issue Blue Book. Space Communications Protocol Specification (SCPS), Transport Protocol (SCPS-TP). Recommendation for Space Data System Standards, 2006.

[SDC+03] Mukundan Sridharan, Arjan Durresi, Sriram Chellappan, Hitay Ozbay, and Raj Jain. Tuning red parameters in satellite networks using control theory. 2003.

[SHJK06] Alexander Sayenko, Timo Hämäläinen, Jyrki Joutsensalo, and Lari Kannisto. Comparison and analysis of the revenue-based adaptive queuing models. *Comput. Netw.*, 50(8):1040–1058, June 2006.

[SHJR04] A. Sayenko, T. Hamalainen, J. Joutsensalo, and P. Raatikainen. Adaptive scheduling using the revenue-based weighted round robin. In *Networks, 2004. (ICON 2004). Proceedings. 12th IEEE International Conference on*, volume 2, pages 743 – 749 vol.2, nov. 2004.

[SSF08] R. Secchi, A. Sathiaseelan, and G. Fairhurst. Scheduling tcp-friendly flows over a satellite network. In *Satellite and Space Communications, 2008. IWSSC 2008. IEEE International Workshop on*, pages 155 –159, oct. 2008.

[TKN06] T. Taleb, N. Kato, and Y. Nemoto. REFWA: An Efficient and Fair Congestion Control Scheme for LEO Satellite Networks. *Networking, IEEE/ACM Transactions on*, 14(5):1031–1044, Oct. 2006.

[TLL07] Shih-Ching Tsao, Yuan-Cheng Lai, and Ying-Dar Lin. Taxonomy and evaluation of tcp-friendly congestion-control schemes on fairness, aggressiveness, and responsiveness. *Network, IEEE*, 21(6):6 –15, november-december 2007.

[TSZS06] K. Tan, J. Song, Q. Zhang, and M. Sridharan. A compound tcp approach for high-speed and long distance networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1 –12, april 2006.

[VF08] Christophe De Vleeschouwer and Pascal Frossard. Loss-resilient window-based congestion control. *Computer Networks*, 52(7):1473 – 1491, 2008.

[VKM02] D. Velenis, D. Kalogeras, and B. Maglaris. SaTPEP: A TCP Performance Enhancing Proxy for Satellite Links. In *NETWORKING 2002: Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, volume 2345 of *Lecture Notes in Computer Science*, pages 1233–1238. Springer-Verlag, 2002.

[WRMG09] Hangxing Wu, Fengyuan Ren, Dejun Mu, and Xianwu Gong. An efficient and fair explicit congestion control protocol for high bandwidth-delay product networks. *Computer Communications*, 32(7-10):1138 – 1147, 2009.

[WWSS01] Haining Wang, Haining Wangy, Chia Shen, and Kang G. Shin. Adaptive-weighted packet scheduling for premium service. In *Communications, 2001. Proceedings. IEEE International Conference on*, volume 6, pages 1846–1850, 2001.

[WZ02] B.P. Wydrowski and M. Zukerman. Maxnet: A congestion control architecture. *IEEE Communications Letters*, 6:512–514, 2002.

[YDKD02] Sungwon Yi, Xidong Deng, G. Kesidis, and C.R. Das. Providing fairness in diffserv architecture. In *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, volume 2, pages 1435 – 1439 vol.2, nov. 2002.

[Yeh02] E. Yeh. Delay-optimal rate allocation in multiaccess communications: a cross-layer view. In *Multimedia Signal Processing, 2002 IEEE Workshop on*, pages 404 – 407, dec. 2002.

[ZLL08] Yueping Zhang, Derek Leonard, and Dmitri Loguinov. Jetmax: Scalable max-min congestion control for high-speed heterogeneous networks. *Computer Networks*, 52(6):1193 – 1219, 2008.

[ZRK06] Jing Zhu, Sumit Roy, and Jae H. Kim. Performance modelling of tcp enhancements in terrestrial-satellite hybrid networks. *IEEE/ACM Trans. Netw.*, 14(4):753–766, 2006.

E. Rendon-Morales Bio

Elizabeth Rendon-Morales has received her BEng (Honors) in Telecommunication Engineering from National University of México and her master's degree in Information Technologies from the Technology Institute of Mexico (ITAM) in 2006. She has obtained a second master's degree in Networks and Multimedia Information Systems from TELECOM Bretagne in Rennes, France. Since 2009, she is currently a PhD candidate at the Technical University of Catalonia (UPC) in Spain. Her research interests are focused on QoS provisioning over wireless system, including transport protocols and cross-layer design.